Titre: Title:	Determinism Enhancement and Reliability Assessment in Safety Critical AFDX Networks
Auteur: Author:	Meng Li
Date:	2016
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Li, M. (2016). Determinism Enhancement and Reliability Assessment in Safety Critical AFDX Networks [Ph.D. thesis, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/2095/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/2095/
Directeurs de recherche: Advisors:	Guchuan Zhu, & Yvon Savaria
Programme: Program:	génie électrique

UNIVERSITÉ DE MONTRÉAL

DETERMINISM ENHANCEMENT AND RELIABILITY ASSESSMENT IN SAFETY CRITICAL AFDX NETWORKS

MENG LI
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIÆ DOCTOR
(GÉNIE ÉLECTRIQUE)
AVRIL 2016

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

DETERMINISM ENHANCEMENT AND RELIABILITY ASSESSMENT IN SAFETY CRITICAL AFDX NETWORKS

présentée par : LI Meng
en vue de l'obtention du diplôme de : Philosophiæ Doctor
a été dûment acceptée par le jury d'examen constitué de :

M. SAUSSIÉ David A., Ph. D., président
M. ZHU Guchuan, Doctorat, membre et directeur de recherche
M. SAVARIA Yvon, Ph. D., membre et codirecteur de recherche
M. MULLINS John, Ph. D., membre
M. ZENG Haibo, Ph. D., membre externe

DEDICATION

I dedicate this work to all my beloved family : my wife, my parents, and my brother, who offered unconditional love and support and have always been there for me. Thanks for believing in me and supporting me to strive for my dreams.

ACKNOWLEDGEMENTS

I would like to thank my supervisors, Professor Guchuan Zhu and Professor Yvon Savaria for all their guidance and encouragement during my doctoral research at École Polytechnique de Montréal. Their continued support as well as the many interesting discussions helped me learn and understand the research methodology, which will definitely benefit my research career in the future.

I would like to thank my committee members, professor David Saussié from the Department of Electrical Engineering, Polytechnique Montréal, professor John Mullins from the Department of Computer Engineering, Polytechnique Montréal, and professor Haibo Zeng from the Department of Electrical & Computer Engineering, Virginia Tech for their time spent reviewing my thesis and attending my defense session, and also for their comments and suggestions.

I sincerely thank all the team members who worked previously in the AVIO402 project, particularly Michaël Lauer, Jian Li, José-Philippe Tremblay, Safwen Bouanen, Romain Nishi, Talal Zakani, Davide Trentin, Federico Montano and Anh Hai David Nguyen. Working with them, I developed a better understanding of AFDX and learnt lots of knowledge regarding related technologies.

Finally, I am also grateful for the financial support from the China Scholarship Council (CSC). Special thanks are due to NSERC-CRIAQ CRD project AVIO402, MITACS Acceleration Quebec program, MDEIE-CRIAQ Quebec-China project, and the industrial partners Thales Canada Inc. and Bombardier Aerospace that have sponsored in part this research.

RÉSUMÉ

AFDX est une technologie basée sur Ethernet, qui a été développée pour répondre aux défis qui découlent du nombre croissant d'applications qui transmettent des données de criticité variable dans les systèmes modernes d'avionique modulaire intégrée (Integrated Modular Avionics). Cette technologie de sécurité critique a été notamment normalisée dans la partie 7 de la norme ARINC 664, dont le but est de définir un réseau déterministe fournissant des garanties de performance prévisibles. En particulier, AFDX est composé de deux réseaux redondants, qui fournissent la haute fiabilité requise pour assurer son déterminisme.

Le déterminisme de AFDX est principalement réalisé par le concept de liens virtuels (Virtual Links), qui définit une connexion unidirectionnelle logique entre les points terminaux (End Systems). Pour les liens virtuels, les limites supérieures des délais de bout en bout peuvent être obtenues en utilisant des approches comme calcul réseau, mieux connu sous l'appellation Network Calculus. Cependant, il a été prouvé que ces limites supérieures sont pessimistes dans de nombreux cas, ce qui peut conduire à une utilisation inefficace des ressources et augmenter la complexité de la conception du réseau. En outre, en raison de l'asynchronisme de leur fonctionnement, il existe plusieurs sources de non-déterminisme dans les réseaux AFDX. Ceci introduit un problème en lien avec la détection des défauts en temps réel. En outre, même si un mécanisme de gestion de la redondance est utilisé pour améliorer la fiabilité des réseaux AFDX, il y a un risque potentiel souligné dans la partie 7 de la norme ARINC 664. La situation citée peut causer une panne en dépit des transmissions redondantes dans certains cas particuliers. Par conséquent, l'objectif de cette thèse est d'améliorer la performance et la fiabilité des réseaux AFDX.

Tout d'abord, un mécanisme fondé sur l'insertion de trames est proposé pour renforcer le déterminisme de l'arrivée des trames au sein des réseaux AFDX. Parce que la charge du réseau et la bande passante moyenne utilisée augmente due à l'insertion de trames, une stratégie d'agrégation des Sub-Virtual Links est introduite et formulée comme un problème d'optimisation multi-objectif. En outre, trois algorithmes ont été développés pour résoudre le problème d'optimisation multi-objectif correspondant. Ensuite, une approche est introduite pour incorporer l'analyse de la performance dans l'évaluation de la fiabilité en considérant les violations des délais comme des pannes. De cette façon, le resserrement des limites supérieures probabilistes pour les liens virtuels (Virtual Links) peuvent être appliquées à la certification des réseaux AFDX. Afin de procéder à l'analyse de la fiabilité, la technique d'analyse des arbres de pannes (FTA) est combinée au calcul des réseaux stochastiques pour calculer les

limites supérieures des délais. Enfin, une analyse mathématique des pannes dans la gestion de la redondance du protocole AFDX est fournie. Afin d'éliminer ce risque de défaillance, un modèle de courbe d'arrivée plus précis est combiné à deux techniques d'optimisation pour réduire la borne supérieure de la gigue temporelle qui limite la fiabilité des réseaux redondants. Afin de valider les performances de toutes les approches proposées, des études de cas sont réalisées et les résultats présentés confirment la faisabilité et l'applicabilité des méthodes proposées.

ABSTRACT

AFDX is an Ethernet-based technology that has been developed to meet the challenges due to the growing number of data-intensive applications in modern Integrated Modular Avionics systems. This safety critical technology has been standardized in ARINC 664 Part 7, whose purpose is to define a deterministic network by providing predictable performance guarantees. In particular, AFDX is composed of two redundant networks, which provide the determinism required to obtain the desired high reliability.

The determinism of AFDX is mainly achieved by the concept of Virtual Link, which defines a logical unidirectional connection from one source End System to one or more destination End Systems. For Virtual Links, the end-to-end delay upper bounds can be obtained by using the Network Calculus. However, it has been proved that such upper bounds are pessimistic in many cases, which may lead to an inefficient use of resources and aggravate network design complexity. Besides, due to asynchronism, there exists a source of non-determinism in AFDX networks, namely frame arrival uncertainty in a destination End System. This issue introduces a problem in terms of real-time fault detection. Furthermore, although a redundancy management mechanism is employed to enhance the reliability of AFDX networks, there still exist potential risks as pointed out in ARINC 664 Part 7, which may fail redundant transmissions in some special cases. Therefore, the purpose of this thesis is to improve the performance and the reliability of AFDX networks.

First, a mechanism based on frame insertion is proposed to enhance the determinism of frame arrival within AFDX networks. As the network load and the average bandwidth used by a Virtual Link increase due to frame insertion, a Sub-Virtual Link aggregation strategy, formulated as a multi-objective optimization problem, is introduced. In addition, three algorithms have been developed to solve the corresponding multi-objective optimization problem. Next, an approach is introduced to incorporate performance analysis into reliability assessment by considering delay violations as failures. This allowed deriving tighter probabilistic upper bounds for Virtual Links that could be applied in AFDX network certification. In order to conduct the necessary reliability analysis, the well-known Fault-Tree Analysis technique is employed and Stochastic Network Calculus is applied to compute the upper bounds with various probability limits. Last, a mathematical analysis of redundancy management failures in the AFDX protocol is provided. In order to eliminate this potential failure, a staircase model is applied to obtain tighter jitter bound estimations and two approaches are proposed and investigated to mitigate the jitter pessimism. In order to validate the performance of all

the proposed approaches, case studies are carried out individually and the reported results confirm their feasibility and applicability.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ADIRU	Air Data Inertial Reference Unit
AFDX	Avionics Full-Duplex Switched Ethernet
AFR	Arrival Frame Rate
ARINC	Aeronautical Radio, Incorporated
ATM	Asynchronous Transfer Mode
BAG	Bandwidth Allocation Gap
BC	Bus Controller
BCD	Binary Coded Decimal
BE	Best-Effort
BM	Bus Monitor
BNR	Binary
CAN	Controller Area Network
CCA	Common Cause Analysis
CM	Compression Master
COTS	Commercial Off-The-Shelf
CRC	Cyclic Redundancy Check
DA	Destination Address
EBB	Exponentially Bounded Burstiness
EBF	Exponentially Bounded Fluctuation
EDF	Earliest Deadline First
ES	End System
FCFS	First-Come, First-Served
FCS	Frame Check Sequence
FIFO	First-In, First-Out
FMEA	Failure Modes and Effects Analysis
FMECA	Failure Modes, Effects, and Criticality Analysis
FP-RR	Fixed Priority combined with Round Robin
FTA	Fault Tree Analysis
FVW	First Valid Wins
IC	Integrity Checking
IFG	Inter Frame Gap
IMA	Integrated Modular Avionics
LRU	Line Replaceable Unit

LS	Local Synchronization
LSB	Least Significant Bit
MAC	Media Access Control
MFS	Maximum Frame Size
MGF	Moment Generating Function
MSB	Most Significant Bit
NC	Network Calculus
OS	Operating System
PBOO	Pay Bursts Only Once
PSN	Previous Sequence Number
PSSA	Preliminary System Safety Analysis
QoS	Quality of Service
RC	Rate-Constrained
RFTR	Required Frame Transmission Rate
RM	Redundancy Management
RPG	Redundant Paths Group
RR	Round-Robin
RT	Remote Terminal
RTC	Remote Terminal Cluster
SA	Source Address
SBB	Stochastically Bounded Burstiness
SC	Synchronization Client
SDI	Source/Destination Identifier
SFCC	Slap Flat Control Computer
SFCS	Slat Flap Control System
SFD	Start Frame Delimiter
SM	Synchronization Master
SN	Sequence Number
SNC	Stochastic Network Calculus
SOF	Start Sf Frame
SSM	Sign/Status Matrix
Sub-VL	Sub-Virtual Link
TLDM	Transmission Latency Difference Minimization
TT	Time-Triggered
TTP	Time Triggered Protocol
VL	Virtual Link

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

Avionics Full-Duplex Switched Ethernet (AFDX) is a promising solution that can ensure deterministic communications for safety critical avionics applications. This technology is standardized in ARINC 664 Part 7. Nevertheless, it is recognized that determinism and reliability are two main concerns for the AFDX network protocol, which form the main research subjects of this thesis. In this chapter, the context of this research and a literature review related to performance analysis and reliability assessment of AFDX networks are introduced. Then, the main contributions of this research are summarized. The organization of this thesis is presented at the end of this chapter.

1.1 Context of the Present Research Project

The introduction of fly-by-wire avionic systems has caused growing concerns in terms of their performance and reliability. With safety critical systems in which failures may be catastrophic, it is essential to guarantee reliable communications among avionic systems at every flight phase. To meet stringent reliability requirements, certain technologies, such as ARINC 429, have been developed and successfully deployed since the late 1970s [53]. However, as the amount of electronic components in aircrafts continues to increase, legacy avionic communication protocols are at their limit in terms of performance, e.g., bandwidth and throughput [111]. Moreover, the interconnection of individual subsystems through physical point-to-point connections requires a large amount of wires, which result in cabling burdens, and increased network complexity that impact the total weight of an aircraft [53]. In addition, stringent safety requirements that impose using redundant components further aggravate the situation. Therefore, new aircraft network technologies have been developed in order to meet the ever-increasing demand in avionic communications, among which Ethernet-based technology is recognized as one of the most promising and dominant solutions. In contrast to standard Ethernet networks, the key focus in avionics is put on safety rather than throughput. Thus, particular adaptations are required to integrate Ethernet-based technologies in safety critical avionic applications.

One of the main obstacles to the application of Ethernet technology in avionic systems lies in Ethernet's non-determinism. For standard Ethernet networks, frame timeout or loss is a common issue. However, these shortcomings are unacceptable for critical real-time applications in aircrafts. Inspired by the concept of asynchronous transfer mode (ATM), a concept of Virtual Link (VL) has been introduced in AFDX to establish a virtual point-

to-point connection between the source and destination End Systems (ESs). Moreover, a maximum bandwidth allocated to this connection is imposed to limit bandwidth utilization. Essentially, two mechanisms are used to ensure that the bounded data transmission rate is respected. In each source ES, traffic shaping is employed to control the flow through each VL in accordance with the so-called Bandwidth Allocation Gap (BAG), which defines the minimum time interval between successive frames in a VL. At the ingress of the switches, traffic policing is used to protect the network from babbling-idiot failures. Furthermore, the routes of the VLs are statically defined off-line, which allows further enhancing determinism. Up to now, much work has been dedicated to evaluate the end-to-end frame transit delay in order to provide a firm, mathematically provable, upper bound.

Other techniques are also applied in AFDX to mitigate the possibility of frame loss or timeout. For instance, full-duplex communication is employed to eliminate the occurrence of frame collisions, which frequently occur in standard Ethernet networks when two devices attempt to simultaneously transmit data on the same physical link. Furthermore, AFDX is composed of two redundant networks to minimize the risk of data loss and thus provide a high reliability.

Nevertheless, there are still some potential problems in AFDX networks in terms of criticality for avionics applications. First, there is a source of non-determinism related to fault detection in the destination ESs. Indeed, the AFDX standard does not force a VL to transmit frames if there is no data to transmit, even though the VL is available. This means that destination ESs cannot detect one or several consecutive frame losses (due to frame corruptions or device malfunctions on both redundant networks) until a valid frame arrives. For safety critical applications, this situation raises a serious concern in terms of determinism and reliability. Second, as pointed out in the ARINC 664-P7 standard (see, Section 3.2.6 in [13]), the redundancy management mechanism in AFDX networks may fail under special conditions. This presents a real challenge to system designers because of the lack of analytical frameworks for this problem.

1.2 Performance Evaluation and Reliability Assessment of Avionics Networks

With the evolution of fly-by-wire technologies, avionic data networks play an essential role of an ever-increasing importance in the control and communication architecture of aircrafts. Thus, the evaluation of such networks is required to ensure their dependability in safety critical applications. In [48], some criteria for evaluating data networks have been developed for a wide range of networks including AFDX. This work considers diverse attributes in terms of performance, safety, and certification. Among the considered attributes, one of the most important characteristics is determinism, which is not only applicable to media

access control (MAC), but also to the behavior of a system. Determinism of a system implies that its behavior should be predictable based on current state, or be determined *a priori*. Another consideration is related to reliability assessment, which can be performed in either a bottom-up or a top-down manner. The available approaches for bottom-up analysis include failure modes and effects analysis (FMEA) and failure modes, effects, and criticality analysis (FMECA). The existing top-down methods are common cause analysis (CCA), preliminary system safety analysis (PSSA), and fault tree analysis (FTA). This thesis addresses mainly determinism issues in the AFDX protocol, aiming at determinism enhancement, performance improvement, and reliability assessment.

1.3 Techniques for Timing Analysis in AFDX Networks

It is pointed out in [13] that *for safety critical systems, reliable "real-time" communication links are essential*. Therefore, in order for AFDX networks to be considered in safety critical applications, it is essential to guarantee that this network can support deterministic communications. This is however very challenging due to the fact that AFDX is an asynchronous protocol. Specifically, frame transmission delay in AFDX networks can be divided into a fixed part and a variable part [109]. The fixed part is the sum of the transmission latencies over physical links and the technological latencies induced by different network elements. The variable part is due to the congestion in the output ports of ESs or switches. It depends highly on the load and the scheduling algorithms employed by the multiplexers. In fact, the main challenge arises from the variable part, i.e., the jitter during transmission. Therefore, much effort has been dedicated to estimate the upper bounds for data transmission in order to guarantee timing behavior of the network based on formal analysis.

1.3.1 Deterministic Network Calculus

Deterministic Network Calculus (NC) is a mathematical tool that has been widely applied in performance analysis of communication networks. It was first introduced for characterization of affine traffics [40, 41], followed by a more detailed formulation as given in [78]. In general, deterministic NC is based on the so-called min-plus algebra. Such a tool allows to gain a deep insight into many fundamental problems arising from network traffic engineering.

The deterministic NC provides an elegant framework for determining upper bounds on delay and backlog (or buffer dimensioning), and it has proven to be a valuable and versatile tool for worst-case performance analysis. Many applications can be found in the analysis of, e.g., Internet [50, 78] or in switched Ethernet [56, 110, 55]. In [78], a principle of "Pay Bursts

Only Once" (PBOO) is proposed based on the property of the convolution of service curves under a concatenation case, which contributes to tightening delay bound estimations. In [50], the principle PBOO is extended to aggregate scheduling schemes, resulting in tighter upper bounds on end-to-end delays. Significant improvements in delay estimation were confirmed by numerical results. In [56] and [110], the multiplexing issue is investigated. In [56], an implementation of a weighted and fair queueing policy based on a weighted round robin scheme is reported. It offers a more balanced access to network outputs, rather than a strict priority policy. A more general assumption in term of arbitrary multiplexing instead of a First-In, First-Out (FIFO) aggregate multiplexing is used in [110]. An optimization-based bounding method is then employed to find tight delay bounds instead of the direct application of deterministic NC.

Deterministic NC has also been applied in many studies related to AFDX network performance analysis, e.g., [24, 51, 82, 88], in which an error-free (no frame loss) point-to-point communication is assumed. Furthermore, in AFDX networks, an input data flow is regulated with a minimum time interval, i.e., the BAG. The burst transmission of each flow is upper bounded by a σ-constraint, which guarantees that the quantity of data in any frame will never exceed a constant value σ. In [24], an aggregation of flows sharing one FIFO scheduler is investigated, considering the principle of PBOO. A method for handling an aggregation globally and individually is given in the case of a FIFO policy. In [82], a scheduling policy, namely the Fixed Priority combined with Round Robin (FP-RR), is associated with deterministic NC to derive a tight delay bound for AFDX networks. Deterministic NC is able to handle both periodic and aperiodic flows, although aperiodic flow is the common assumption used in most applications. In [88], periodic flows with known offsets in source ESs are considered to eliminate some pessimistic scenarios, and then the integration of offsets in deterministic NC is investigated. This extended approach in the presence of offsets is evaluated on an industrial AFDX configuration, and the results show that the upper delay bound estimation can be significantly improved.

1.3.2 Stochastic Network Calculus

In recent years, there has been much interest in stochastic extensions of NC. This is motivated by the fact that although deterministic NC provides safe upper bounds for the safety critical applications, the obtained delay bounds are pessimistic. Hence, the overestimation of delay upper bounds leads to inefficient utilization of network resources on average. Unlike deterministic analysis, stochastic network calculus (SNC) provides a means to improve performance bound estimations by capturing the probabilistic nature of system behavior. In

general, the stochastic performance metric can be expressed as

$$\Pr\{\text{Performance is worse than a certain bound}\} \leq \varepsilon,$$

where ε is the admissible violation probability.

Early work dealing with SNC can be found in [75] and [31]. The calculation of stochastic bounds is presented in [75] based on the characterized arrival process in certain intervals. In [31] a $(\rho(\theta), \sigma(\theta))$ model, namely envelope process with respect to θ, is proposed based on the moment generating function (MGF), which is further detailed in [68]. Since then, numerous statistical models for the arrival curve have been developed including Exponentially Bounded Burstiness (EBB) [131], Stochastically Bounded Burstiness (SBB) [118], and Effective Envelope [81]. In [34] and [68], the relationship and differences between different arrival models are detailed. Besides, the stochastic service curve is also investigated under the contention-based multi-access protocols, e.g., CSMA/CD in Ethernet. Similar to the arrival curve, diverse stochastic models are introduced to characterize the service curve, among which we can find Exponentially Bounded Fluctuation (EBF) service curve [79], effective service curve [28], statistical leftover service curve [35], and service curve based on MGFs [129]. By applying the statistical models of arrivals and/or servers, tight performance bounds can be produced by applying the SNC.

An alternative approach is investigated to obtain probabilistic performance bounds concerning independent individually regulated flows with a constant service curve [73, 32, 124, 125]. This approach takes into account the statistical multiplexing gain property, which is indeed an application of the Hoeffding's inequality [125]. In both [73] and [32], the Chernoff bound was applied to derive the stochastic bound under a general traffic constraint. The bound given in [32] is proven to be asymptotically tighter than the one proposed in [73]. Both sets of results are further extended to more practical cases, e.g., heterogeneous regulated inputs, in [125]. Furthermore, the result in [32] with respect to the homogeneous case is slightly improved in [125]. In [124], a better estimate for the heterogeneous cases was obtained by applying a super-additive service curve in [125]. The application of stochastic bounds in the analysis of AFDX networks can be found in [109], where the best result presented in [124] was applied.

1.3.3 Trajectory Approach

In contrast to NC, which considers the worst-case performance on each node, the trajectory approach takes into account the worst-case scenario experienced by a frame along its path.

This technique is first introduced in [93] based on a fixed priority scheduling and then has been elaborated in [92] for schedulability analysis based on a FIFO policy.

This approach has then been applied to AFDX networks in [19, 20] based on the FIFO policy. In these studies, a grouping technique (or serialization effect) is taken into account to mitigate the pessimism. A similar result is reported in [72]. Another application of the trajectory approach to the analysis of AFDX networks can also be found in [89], in which the source of pessimism in the computation of upper bounds with the trajectory approach is characterized in a general formulation. A serialization correction is further developed in [87] to obtain a tighter delay upper bound.

The trajectory approach can provide more deterministic and quantitative guarantees by eliminating impossible scenarios. However, the computation of exact worst-case end-to-end delays is hard to achieve. In fact, the upper bounds obtained with the trajectory approach are still pessimistic in many cases. Further studies are required to tackle the remaining challenges in order to consider the sources of pessimism within a network.

1.3.4 Simulation Approach

The simulation approach aims at imitating the behaviors of the real systems and then provides the information on the end-to-end delay distribution. As on a given set of scenarios, the simulation approach produces more realistic results compared with the NC or trajectory approaches. It can also be applied to evaluate the pessimism of the computed upper bounds. This approach has been investigated for the timing analysis of AFDX [108, 33, 109, 130, 119, 116]. A main challenge related to the simulation approach lies on how to find a representative subset among a huge number of possible scenarios and how to properly define a simulation model that allows holding the characteristics of the practical system. In [108], an approach is proposed to solve the aforementioned problem by classifying the VLs based on their influence. Furthermore, for better imitating practical systems, software packages for real-time simulation, e.g., TrueTime [119], were introduced to perform the analysis. These simulators are able to cope with different scheduling policies.

The simulation approach facilitates the certification of scheduling process. There is no need to wait for the implementation of a real system to test different scheduling policies, which can be implemented on simulation system and verified in advance. Obtaining the distribution of the end-to-end delay is helpful for prototyping the whole system with respect to the configuration. However, the significant drawback of the simulation approach is that the worst-case scenarios, which are rare events, may not be captured in the simulations. Therefore, there is no guarantee that the experimental upper bounds are the exact worst-case delay

bounds.

1.3.5 Model Checking

Model checking is a verification technique based on modeling system behavior in a mathematically precise and unambiguous manner [17]. This approach can help determine an exact worst case end-to-end delay and the corresponding scenario since it explores all the possible states of the modeled system. System modeling can be performed with different formalisms with respect to the system properties. As for time-critical systems, the timed automata is developed by imposing the timing constraints [11]. Applications of this tool to the analysis of AFDX networks can be found in [12, 33, 5, 6, 7]. However, due to the combinatorial explosion problem, the timed automata-based approach cannot cope with large network configurations. For example, in [7], the proposed approach allows for the analysis of systems comprising up to 18 flows, and the admissible configuration is only extended up to 20 from 10 in [6], even with a drastic reduction of the number of scenarios. Thus, it is a real challenge to apply model checking on a practical industrial network configuration.

1.4 Research Contributions

Based on the previous discussion about AFDX networks and the related techniques, the subject of this thesis focuses on further enhancing network determinism, performing the reliability assessment incorporating performance analysis, and eliminating the potential failure within the AFDX protocol.

In this thesis, a mechanism based on frame insertion is first proposed, allowing the enhancement of data transmission determinism in AFDX networks. Indeed, the AFDX standard does not force a VL to transmit frames if there is no data for transmission, even though the VL is eligible. This means that the destination ESs cannot detect frame losses (due to frame corruptions or device malfunctions on both redundant networks) until a correct frame arrives. For critical applications, this raises a serious issue in terms of determinism and reliability. The solution proposed in this thesis is based on the idea of inserting filler frames into VL when its source is silent. This allows the destination ESs of the VL to detect the fault if a frame is missing from the periodical pattern obtained with filler frames. Obviously, this mechanism does not affect the maximum bandwidth reserved for VL. However, inserting frames will increase the network load and the average bandwidth used by a VL. For this reason, a feature described in the AFDX standard, namely Sub-Virtual Link (Sub-VL) aggregation, is leveraged to minimize the impact on overall network performance. This aggregation strat-

egy is then formulated as a multi-objective optimization problem considering the trade-off between load increase and the delay introduced by Sub-VL aggregation. Three algorithms are proposed and investigated to solve the Sub-VL aggregation optimization problem. Simulations are carried out to illustrate the feasibility of the proposed frame insertion mechanism and to validate the performance of the developed algorithms. The results show that the load increase can be dramatically reduced and the delay introduced by Sub-VL aggregation can be mitigated.

As a continuous effort of the previous work, an approach is further proposed to incorporate the performance analysis into reliability assessment for AFDX networks, in which the end-to-end data transit delay violation is modeled as a failure. It is known that the performance analysis with deterministic approaches leads in general to pessimistic delay upper bounds and does not consider the capability of redundant data transmission mechanisms in AFDX networks, which can tolerate certain faults including single path delay violations. The approach developed in this thesis provides the means for evaluating the system performance with tighter delay bounds by exploring the fault tolerance capability of redundant mechanism. Moreover, SNC is applied to compute the upper bounds with various probability limits, which are suitable to support both quantitative and qualitative reliability assessment for AFDX networks.

In the study of the previously addressed issues, all the analyses are based on the assumption that lost or corrupted frames on one network will have no impact on the overall networks due to the redundancy mechanisms provided by the AFDX protocol. However, as stated in the standard ARINC 664-P7, there still exists a potential problem, which may fail redundant transmissions due to sequence inversion in the redundant channels. In this thesis, this phenomenon is explored and a mathematical analysis is provided. It is revealed that the variable jitter and the transmission latency difference between two successive frames are the two main sources of sequence inversion. Thus, on one hand, a staircase model is applied to characterize the arrival curve in order to obtain tighter jitter bound estimations. On the other hand, two methods are proposed and investigated to mitigate the jitter pessimism, which can eliminate the potential risk.

The main contributions of this thesis can be summarized below.

1. A mechanism based on frame insertion is developed to enhance the determinism of the network. This mechanism allows for real time fault detection in destination ESs. Furthermore, a Sub-VL aggregation strategy is proposed to mitigate the network load increase due to frame insertion while simultaneously minimizing the delay introduced by Sub-VL aggregation.

   The results regarding this contribution are presented in the following paper published

in *IEEE Transactions on Industrial Informatics* [83]:
Meng Li, Michaël Lauer, Guchuan Zhu, and Yvon Savaria. Determinism Enhancement of AFDX Networks via Frame Insertion and Sub-Virtual Link Aggregation. *IEEE Transactions on Industrial Informatics*, vol.10, no.3, pp.1684-1695, Aug. 2014.

2. An approach is introduced to incorporate performance analysis into reliability assessment by considering the delay violation as a failure and establishing a corresponding reliability assessment model. Furthermore, a means is provided for specifying the performance requirements based on tighter bounds associated with probability budgets in order to explore the fault tolerance capabilities of redundant mechanisms.

   The results regarding this contribution are presented in the following paper submitted to *Reliability Engineering & System Safety* [84]:
   Meng Li, Guchuan Zhu, Michaël Lauer, Yvon Savaria, and Jian Li. Incorporating Performance Analysis into Reliability Assessment for Avionics Full-Duplex Switched Ethernet Networks. *Reliability Engineering & System Safety*.

3. A formal analysis is carried out against the potential failures in redundancy management (RM) and the sources of sequence inversion are identified. In order to prevent the phenomenon of sequence inversion from occurring, a staircase model is applied to characterize the arrival curve in order to obtain tighter jitter bound estimates. Furthermore, two methods are proposed and investigated for eliminating the potential failures due to sequence inversion of the redundant networks.

   The results regarding this contribution are presented in the following paper submitted to *IEEE Transactions on Industrial Informatics* [85]:
   Meng Li, Guchuan Zhu, Yvon Savaria, and Michaël Lauer. Reliability Enhancement of Redundancy Management in AFDX Networks. *IEEE Transactions on Industrial Informatics*.

For all the three papers, the doctoral candidate, Meng Li, is the leading author who has made the most substantial contributions. The co-author, Michaël Lauer, was a postdoctoral fellow working together with the candidate at École Polytechnique de Montréal and is now an Assistant Professor in the dependable computing and fault tolerance group at the laboratory for analysis and architecture of systems (LAAS-CNRS), Université de Toulouse. The co-author, Jian Li, is a former research assistant in our group at École Polytechnique de Montréal and is now an Associate Professor of the School of Software at Shanghai Jiao Tong University.

1.5 Thesis Organization

The organization of this thesis is described as follows.

Chapter 2 introduces the evolution of avionics networks and some of the most employed avionics communication network protocols, i.e., ARINC 429, MIL-STD-1553B, ARINC 825, TTEthernet, and AFDX. A comparison is further performed among the presented networks.

Chapter 3 introduces the details of the analysis tools, namely deterministic NC and SNC, which have been applied in AFDX network performance analysis. Then, approaches for multi-objective design optimization are presented. Finally, the FTA, a tool for quantitative reliability assessment, is detailed. These tools are employed in the subsequent chapters.

Chapter 4 presents first the proposed mechanism for determinism enhancement of AFDX networks via frame insertion. Then a feature of the AFDX network, Sub-VL aggregation, is employed to mitigate load increase due to frame insertion, which has been formulated as a multi-objective optimization problem by considering the trade-off between traffic load and delay due to Sub-VL aggregation. Three algorithms have been developed to find solutions to the optimization problem. Experiments are carried out to verify the proposed mechanism. A real-time systems simulation software, TrueTime, has been utilized to validate the proposed mechanism considering Sub-VL aggregation.

Chapter 5 first introduces an approach in which the end-to-end delay violation is modeled as a failure so that performance analysis can be incorporated into the overall system reliability assessment. Then, the well-known FTA technique is employed to perform reliability assessment while taking into account the failures due to delay violations. SNC is also applied to compute the upper bounds with various probability limits. This approach is illustrated with a case study, and the results confirm that the overall system reliability requirement can be met with less pessimistic probabilistic performance constraints.

Chapter 6 focuses on the phenomenon of sequence inversion, which may induce failures of the redundant transmission in AFDX networks. A mathematical analysis is provided with conditions on the occurrence of this phenomenon. The main sources leading to sequence inversion are due to the jitter and the transmission latency difference between two successive frames. Several solutions that allow avoiding the occurrence of sequence inversion have been developed. The proposed approaches are illustrated through an AFDX network example. Chapter 7 provides a general discussion about the present work, which has been detailed in Chapter 4, Chapter 5, and Chapter 6. Chapter 8 summarizes this thesis and proposes some directions of future work.

Finally, Appendix A introduces the simulation platform, TrueTime, which has been utilized for the validation of the frame insertion mechanism proposed in Chapter 4.

CHAPTER 2 AN OVERVIEW OF AVIONIC NETWORKS

In this chapter, the technological evolution of avionic networks is briefly introduced. Furthermore, several existing avionic protocols are summarized, and a comparison of these network protocols is presented.

2.1 Evolution of Avionic Networks

2.1.1 Technological Evolution

With the emergence of the fly-by-wire concept, avionic networks have attracted an increasing attention due to the growing demand for higher performance, reliability, and safety concerning data communications in avionic systems. Most of the early avionic data buses were essentially digital interfaces allowing the components in an avionic system to be connected together. Among the most popular avionic data bus standards, one can find ARINC 429 for civil aircrafts and MIL-STD-1553B for military systems [117, 38]. Because of their widespread use, these two standards still remain media of choice for many upgrades and improvement products in the avionic industry. Meanwhile, an important effort has also been dedicated to the development of extended and modified versions of these standards in order to meet the performance requirements for modern avionic systems [69, 103]. There is another set of data bus standards originally developed for the automobile industry, but that are now finding their way into aerospace systems, such as the Controller Area Network (CAN), the Time Triggered Protocol (TTP), and FlexRay [61]. Driven by the increasing amount of information flow due to the transmission of premium traffic (e.g., audio and video signals), inter-system communications, and the use of general purpose operating systems, there is a growing need for data networks inside aircrafts. Avionic data transmission systems have evolved from instrumentation oriented digital interfaces to information centric network.

The evolution of design concepts obviously requires new technologies, which can provide high speed, high reliability, and low cost networks. Consequently, there is a trend in aerospace industry to use commercial off-the-shelf (COTS) network technologies in avionics. One of the viable candidates is the high speed IEEE802.3 Ethernet network standard, which has attracted much attention from the aerospace industry [53]. However, the inherent non-determinism of the standard Ethernet prevents a direct utilization of such a technology in safety critical avionic systems. Thus, it is recognized that enhancements and improvements aiming at extending the standard Ethernet are indispensable in order to meet the require-

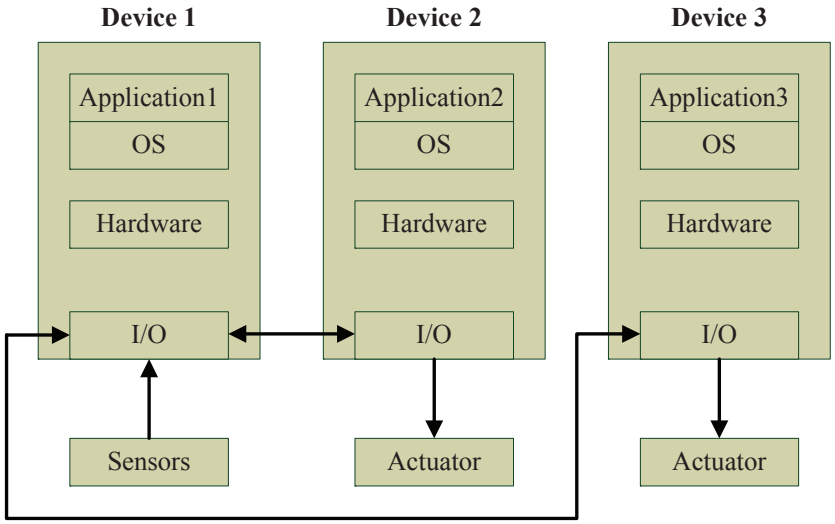
ments in an aerospace environment. In this context, the AFDX protocol has been developed and deployed in Airbus' A380 and is further used in A350, A400M, B787 etc. Another variant of such protocols based on IEEE 802.3 Ethernet is the TTEthernet, in which emphasis is put on enhancing the determinism by imposing strict timing constraints while allowing the integration of time-triggered messages with event-triggered messages on one Ethernet network. In addition, CAN-based technology is also developed to achieve a high level of determinism. This solution has been standardized in ARINC 825 for use in aircrafts, e.g., B787 Dreamliner, which is again a widespread COTS data bus technology. These aforementioned protocols are the preferred technologies considered in the new generation of avionic communication systems.

2.1.2 General Architectures of Avionic Networks

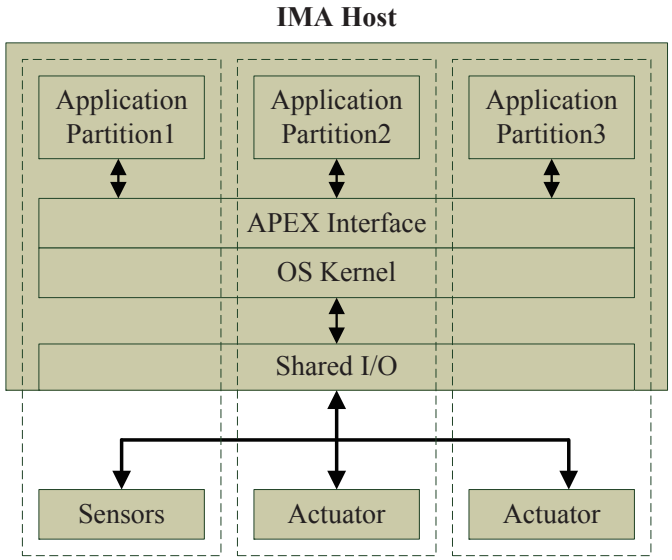
The traditional architecture of avionic networks has been designed in accordance with the federated architecture of avionic systems, in which each function runs on dedicated equipment. This federated architecture allows for a separation of the implementation of different sub-systems and facilitates verification, as limited resource is shared among sub-systems and the dependencies between functions are well understood [43, 23]. However, the federated architecture is not suitable for modern avionic systems due to the increasing functional complexity, because an ever growing number of auxiliary functions requires more installed equipments, more spare parts, higher cost for maintenance, etc. For this reason, there is an increasing need for new architectural paradigms capable of handling stringent requirements of cost savings, flexibility, extensibility, and interoperability with increased functional complexity [43].

In order to meet these challenges, an architecture for new avionic systems, namely the Integrated Modular Avionics (IMA) architecture, has emerged. IMA is very popular in the avionic industry. In sharp contrast to the traditional federated architecture, the IMA architecture enables resource sharing for computing and communications. Thus multiple functions, with possibly different criticality levels corresponding to different sub-systems, can be supported by a common computational platform. With this new technology, the weight, the volume, and the cost of avionic systems can be drastically reduced [23]. Furthermore, in order to prevent interference between functions and the propagation of functional failures, the ARINC 651 standard for IMA design and the ARINC 653 standard for its underling operating system and the software interface specifications, called Application/Executive (APEX), impose a partitioning paradigm. The partition in both space and time domains offers a safe functional isolation mechanism. Therefore, it is possible to add new partitions associated

with additional functions without affecting the already certified modules, as functional isolation is guaranteed. Obviously, the IMA architecture will not only facilitate software design and implementation, but it will also simplify software validation and verification [10]. These features are encouraging the transition from federated avionic architectures to IMA architectures. A comparison of federated architecture and IMA architecture is given in Figure 2.1.



(a) Federated network architecture



(b) IMA architecture

Figure 2.1 A comparison of the federated network and IMA architectures [53, 39].

Nevertheless, the legacy networking standards, such as ARINC 429, cannot meet the requirements of IMA in terms of bandwidth, flexibility, and logical abstraction of networking.

Due to the stringent performance requirements, several new network technologies have been developed, among which we can find AFDX. AFDX was chosen as one solution capable of supporting the IMA architecture, as it can provide high bandwidth communications in a time-deterministic manner. In practice, an AFDX network must interact with other networking protocols to provide feasible and cost-effective solutions for avionic applications. Indeed, an AFDX network can interact with ancillary networks, e.g., ARINC 825, via a gateway to link sensors, actuators or other components.

An example of IMA architecture developed in a research project [120] is shown in Figure 2.2, in which the AFDX is the backbone of the network and all sub-systems are connected via AFDX ESs. Sensors and actuators, as well as Line Replaceable Units (LRUs), geographically close to each other form remote terminal clusters (RTCs). The components in a RTC are connected to an ancillary network, and communicate with the AFDX network via a gateway, composed of a data concentrator and an AFDX ES, through which the data flows over the field busses cross network boundaries (domains). In this example, ARINC 825 is used as the field bus due to its easy connection and configuration flexibility. Cross-domain communication is accomplished by logical communication channels, individual station addressing capabilities, and one-to-many/peer-to-peer communication mechanisms supported by both ARINC 825 and AFDX.

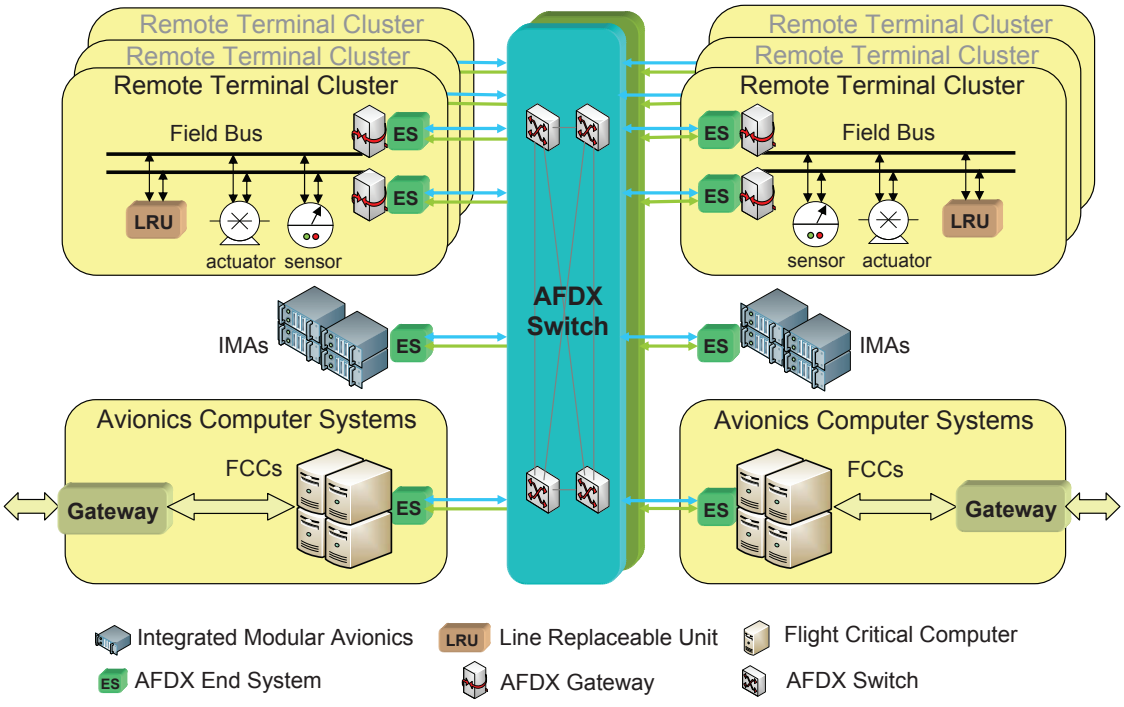


Figure 2.2 An IMA architecture based on an AFDX backbone.

In the following sections of this chapter, we introduce various protocols such as ARINC 429, MIL-STD-1553B, ARINC 825, TTEthernet, and AFDX, which are among the most popular avionic network standards used in the avionic industry. A comparison of the presented avionic network protocols is performed at the end of this chapter.

2.2 ARINC 429

2.2.1 ARINC 429 Protocol

ARINC 429 was first released in 1977 and has since then been widely applied mainly in commercial transport airplanes. Its specification defines notably the word structure for digital data transmission between avionic systems elements. The main characteristics of ARINC 429 are:

— fixed transmission speed at either 12.5 KHz or 100 KHz is predefined for data transmission;

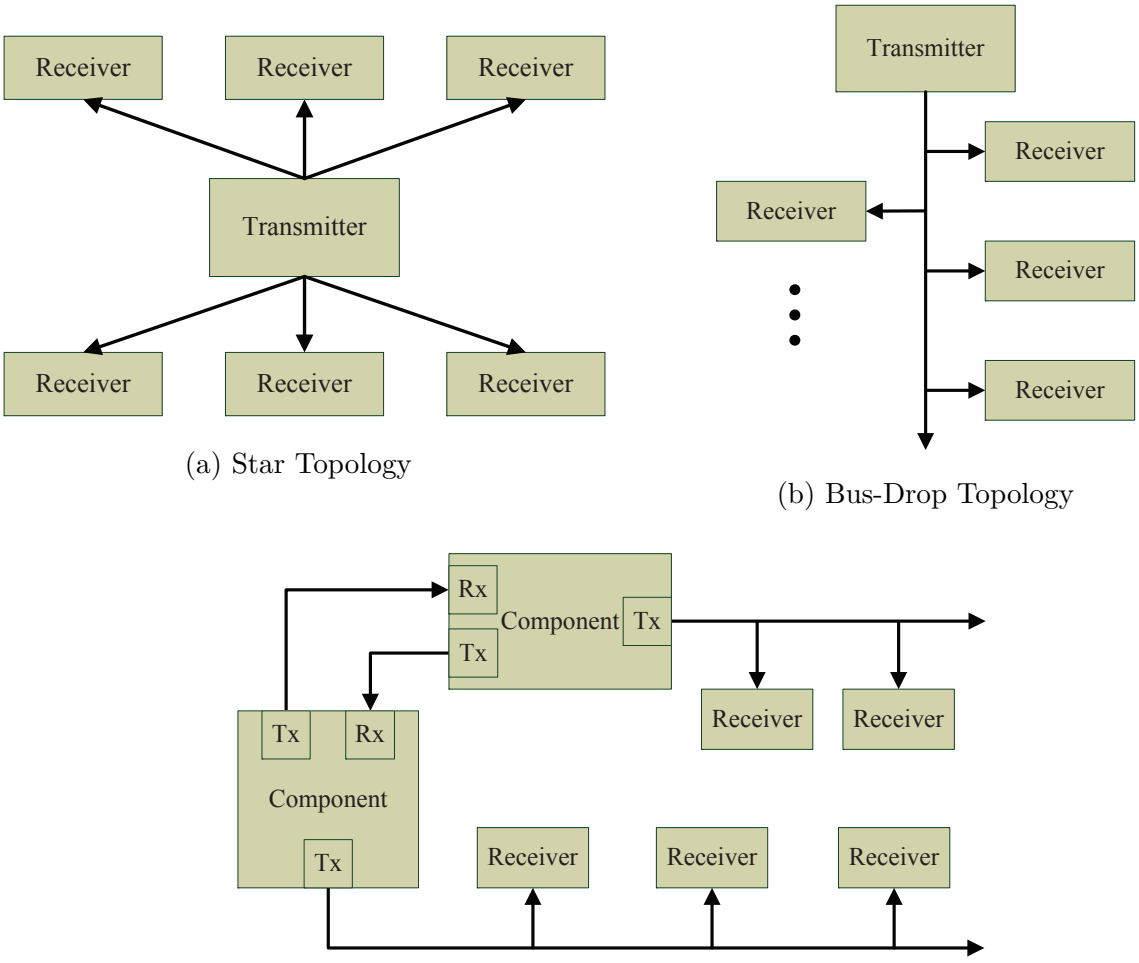
— constant packet size to carry different types of data;

— dedicated point-to-point connections that offer a fixed latency for data packet delivery.

Transmission between components under ARINC 429 is defined as a unidirectional interconnection. The hardware normally consists of a single transmitter and one or up to 20 receivers, which are connected via a twisted and shielded pair of wires. Moreover, the transmitter and the receivers are configured as a star topology, a bus-drop topology or a multiple bus design as shown in Figure 2.3 [52]. As the communication is one way only, each transmitter (denoted by Tx) is 'speak only' and the receiver (denoted by Rx) does not acknowledge for data reception. Due to the unidirectional nature, a device may contain multiple transmitters and receivers via different buses in order to communicate with different components. Thus, fault isolation provisions are performed in each transmitter/receiver to guarantee that the occurrence of failures in either a transmitter or a receiver does not cause any failure to other transmitters/receivers. As a consequence, this simple architecture provides a highly reliable data transmission mechanism for avionic applications.

2.2.2 Word Format of ARINC 429

The basic element in ARINC 429 protocol is a 32 bit digital word made up of five primary fields, which are defined as Parity, Sign/Status Matrix (SSM), Data, Source/Destination Identifier (SDI), and Label [52]. Note that the orders of the most significant bit (MSB) and the least significant bit (LSB) in the Data and Label fields are different. The details of ARINC 429 word format are shown in Figure 2.4.



(a) Star Topology

(b) Bus-Drop Topology

(c) Multiple Bus Design

Figure 2.3 Basic ARINC 429 topologies.

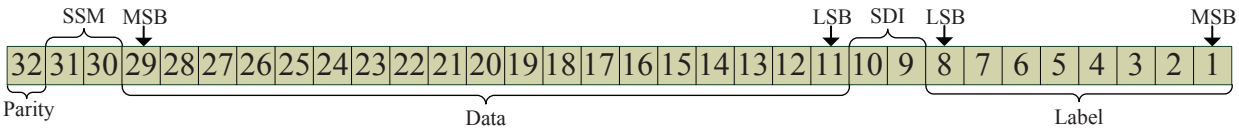


Figure 2.4 The word format of ARINC 429 protocol.

During transmission, Labels are required to identify the data type and the word application. Then the SDI follows to provide the information of source or destination, which is optional under the ARINC 429 specification. The Data field carries information in one of five types: Binary (BNR), Binary Coded Decimal (BCD), Discrete Data, Maintenance Data and Acknowledgement, and Williamsburg/Buckhorn Protocol for file transfers. The availability of diverse data types enables flexibility for practical applications. Subsequently, SSM

cooperates with Label field, and it can provide distinct information for different data types. Furthermore, one parity bit (Bit 32) is utilized in order to complete the transmission with error detection capability.

2.3 MIL-STD-1553B

2.3.1 Overview of MIL-STD-1553B

MIL-STD-1553B, also known as Digital Time Division Command/Response Multiplex Data Bus, is a specification applied in military aircrafts. In general, the MIL-STD-1553B standard defines a redundant, bi-directional, time division multiplexed, semi duplex serial communication standard for avionic systems [100]. Similar to ARINC 429, the frame of MIL-STD-1553B also has a fixed configuration and the transmission bit rate is predefined to be 1 Mbps.

According to the MIL-STD-1553B standard, three types of terminals can be connected to interface with cables as shown in Figure 2.5. They are Bus Controller (BC), Remote Terminal (RT), and Bus Monitor (BM), respectively.

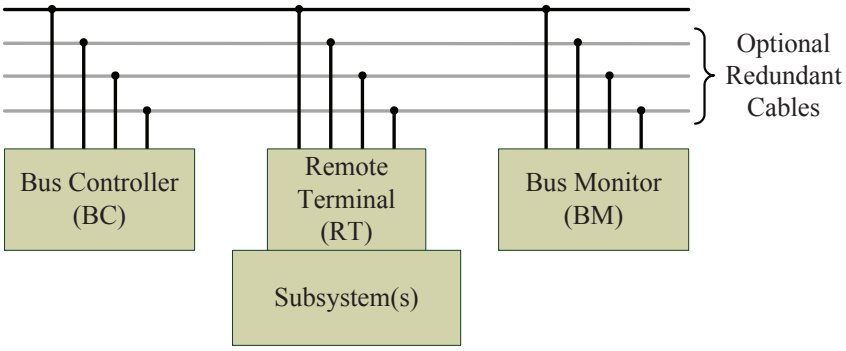


Figure 2.5 MIL-STD-1553B bus architecture.

In general, BCs initiate all transmissions on the bus and all data flows are transmitted under a command/response mode. On the same bus, there may be more than one BC, however only one BC is allowed to be active at any time. A BC sends a command to one or more RTs, which responds accordingly. The RT is designed to establish connections between subsystems and the MIL-STD-1553B data bus. It may be an independent device or be embedded within the subsystem. One data bus can support up to 31 RTs, and each of them is able to communicate with 30 subsystems. The BM is a device listening to the bus traffic and record selected information for post analysis. The primary function of the BM is for system debugging and testing.

2.3.2 Word Formats of MIL-STD-1553B

To establish communications between the master and the slaves, there exist three types of words: command words, data words, and status words. All the frames of MIL-STD-1553B consist of 20 bits in total including three sync bits and one parity bit as shown in Figure 2.6.

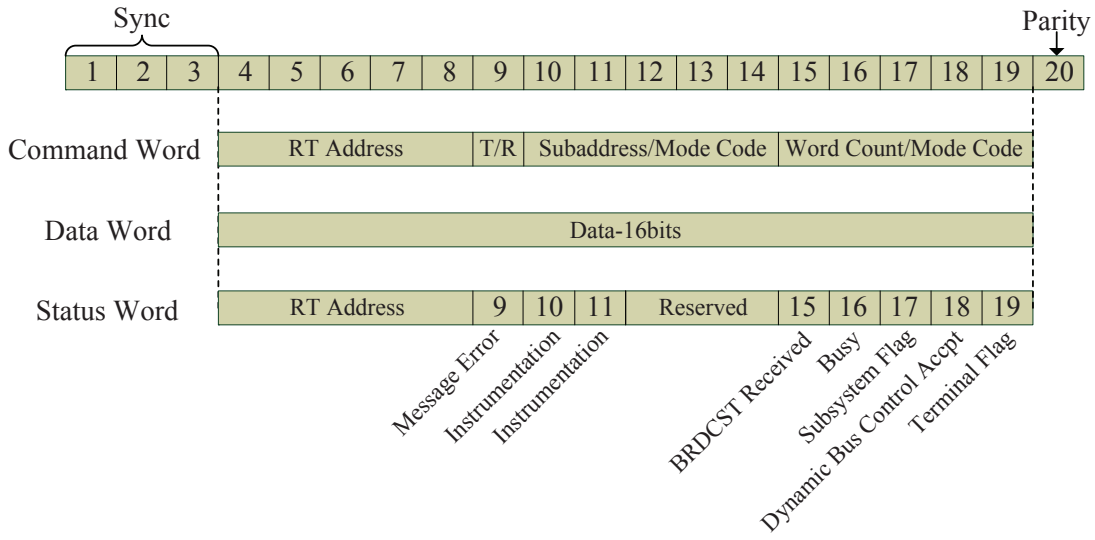


Figure 2.6 The word formats of MIL-STD-1553B.

A command word precedes every data transition. It designates many fields, e.g., data direction and subaddress, for the following operations. Except for broadcast commands, the RT must send a status word as a response to the BC command. If data information is required, the data word is transmitted between BC and RTs or among RTs.

2.4 ARINC 825

2.4.1 Overview of ARINC 825

ARINC 825 is a specification adopted to standardize CAN for aviation applications [14]. The latest version of the CAN specification is CAN 2.0, which is composed of two parts, part A and part B. ARINC 825 adopts CAN 2.0B, which employs the extended frames with 29-bit identifiers. Furthermore, ARINC 825 is intended to support inter-network communications. A gateway is also specified for interfacing with other networks, which indeed allows implementing the concept of remote data concentrator. Typically, the gateway forwards the received data from CAN nodes to a high speed network, e.g. AFDX. Instead of using dedicated communications or master/slave mode, the CAN nodes share a common data bus in

a multi-master manner. Thus, these nodes communicate in a half duplex mode as a result of bus sharing. The raw data rate of CAN bus can reach up to 1Mbps with the cable length shorter than 40 meters. There are four more options for the data rates: 500Kbps, 250Kbps, 125Kbps, and 83.33Kbps. For each rate, there are different constraints on both length and number of attached nodes, which are given in Table 2.1.

Table 2.1 Constraints for different data rates

Data Rate (Kbps)	83.33	125	250	500	1000
Maximum wiring length (m)	200+	160+	80+	80	40
Maximum number of nodes	60	50	40	35	30

Since ARINC 825 is a multi-master network, it is possible that multiple nodes submit messages on the shared bus simultaneously. Thus, an arbitration is required to deal with this problem and authorize the data frame with highest priority to be sent. CAN provides a solution based on a bit-to-bit arbitration, in which the dominant bit (bit 0) supersedes the recessive bit (bit 1). An example is given in Figure 2.7, in which Node 3 wins the arbitration. As shown in this figure, after the start of frame (SOF) bit, each bit is processed following the arbitration rule. All nodes losing the contention stop transmission and switch to listening mode. In the next interframe space, the frames pending for transmission will be automatically retried.

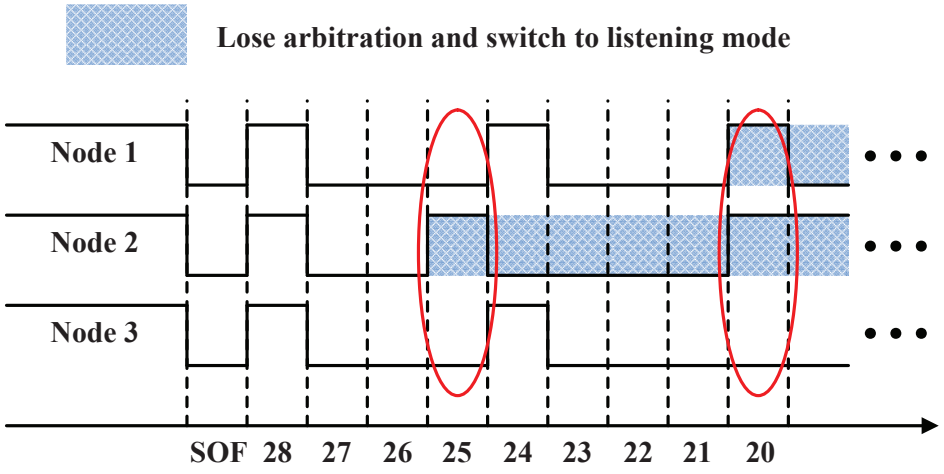


Figure 2.7 ARINC 825 bus arbitration [14].

2.4.2 Data Frame Structure of ARINC 825

Four types of frames are defined in the CAN protocol, namely Error Frame, Remote Frame, Overload Frame, and Data Frame, respectively. For the purpose of comparison, the focus is put on the data frame structure. The data frame consists of a payload up to 8 bytes and an overhead of 8 bytes, including Cyclic Redundancy Check (CRC) fields, as shown in Figure 2.8 [14].

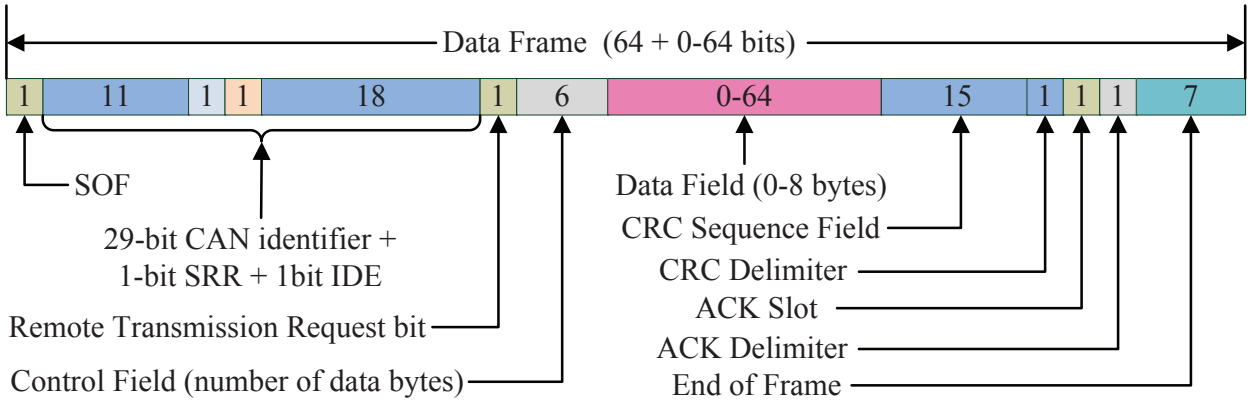


Figure 2.8 ARINC 825 frame structure.

ARINC 825 inherits the broadcast communication mechanism from the CAN standard and defines additional functions to support the peer-to-peer communication mode. These two transmission modes are distinguished by using different format of 29-bit identifier. Typically, a data message carries between 0 and up to 8 bytes of payload, which consists of one or more of the following types: signed integer, unsigned integer, floating-point, enumerated, ASCII, and opaque. The data field is followed by a CRC checksum field, which enhances the capability of error detection compared with ARINC 429 or MIL-STD-1553B.

2.5 TTEthernet

2.5.1 Overview of TTEthernet

TTEthernet, also known as Deterministic Ethernet standardized by SAE AS6802, is a time-triggered Ethernet protocol that extends classical Ethernet for safety-critical and real-time applications [104, 99]. Typically, a TTEthernet network is composed of three elements: End Systems (ESs), switches, and bi-directional physical links. TTEthernet is a synchronized protocol, where determinism is achieved by the timing throughout the system. Therefore, a

global timing is established and maintained within the network. In the TTEthernet protocol, three components are defined to realize the synchronization, which are Synchronization Master (SM), Synchronization Client (SC), and Compression Master (CM), respectively. The synchronization approach is depicted in Figure 2.9. As shown in the figure, in the first step, all the SMs simultaneously send protocol control frames to the CM, and then the CM calculates an average value based on the frame arrival times. In the second step, new protocol control frames are sent to all SMs and the SCs. The designation of a SM or a CM is based on the system architecture. Either an ES or a switch can be configured as a SM, a CM, or a SC. In order to enhance the fault-tolerance capability, a redundancy mechanism is employed in TTEthernet network.

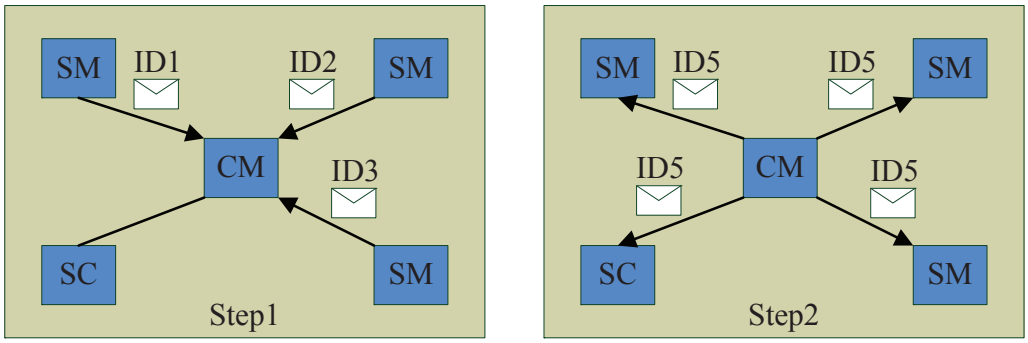


Figure 2.9 TTEthernet synchronization approach.

2.5.2 Frame Classification and Frame Structure of TTEthernet

TTEthernet classifies the traffic into three categories: time-triggered (TT) traffic, rate-constrained (RC) traffic, and best-effort (BE) traffic as shown in Figure 2.10.

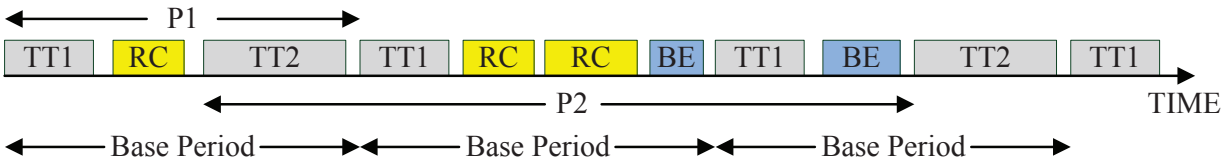


Figure 2.10 TTEthernet traffic classification and frame transmission illustration.

All TT frames are delivered at predefined time slots, which are reserved only for TT traffic communication, and thus these frames have tight jitter. RC frames are transmitted with respect to predefined bandwidth allocations. Thus, successive frames belonging to the same

RC dataflow are regulated based on a minimum time interval to guarantee the bandwidth allocation. It is possible that RC frames from different sources conflict in either switches or destination ESs. Consequently, transmission jitters may be increased due to congestion. However, transmission jitters are upper bounded and deterministic as sufficient bandwidth is allocated in advance for RC traffic. For the BE frames, there is no guarantee for the transmission as this kind of traffic has the lowest priority among the three types. If the bandwidth is occupied by either TT traffic or RC traffic, the BE frames will be delayed until the network is available.

As TTEthernet is fully compliant with IEEE 802.3, the frame structure follows the standard Ethernet frame as shown in Figure 2.11.

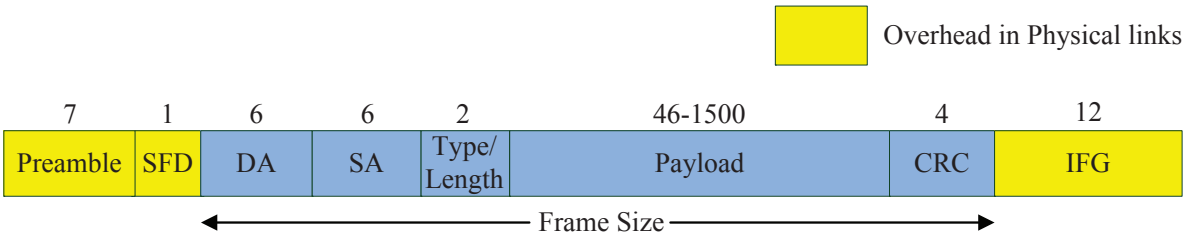


Figure 2.11 TTEthernet frame structure.

The given numbers corresponding to each field indicate the size in bytes. The transmission of an Ethernet frame starts with a preamble field followed by a start frame delimiter (SFD) in physical links. Then the destination address (DA) and the source address (SA) are specified for frame addressing. The address fields are followed by two bytes reserved as type or length indicator. The payload is then followed, which is in the range of 46 to 1500 bytes. At the end of the frame, a 32-bit CRC is specified for error control. Finally an inter frame gap (IFG) is added to guarantee the minimum interval between two successive frames.

2.6 AFDX Networks

AFDX has been developed for supporting safety-critical applications based on the IEEE 802.3 Ethernet protocol. It is a deterministic, redundant, full duplex, and switched network. The determinism is mainly achieved by the concept of Virtual Link (VL). Moreover, the reliability is improved by adopting a redundancy mechanism. The full duplex communication mode adopted in AFDX allows avoiding collisions, which helps further ensure deterministic timing performance. With these key features, AFDX can provide deterministic, reliable networks with guaranteed Quality of Service (QoS) and performance [13].

2.6.1 Overview of AFDX

Typically, an AFDX network is composed of three elements:

— End Systems
— Switches
— Physical links

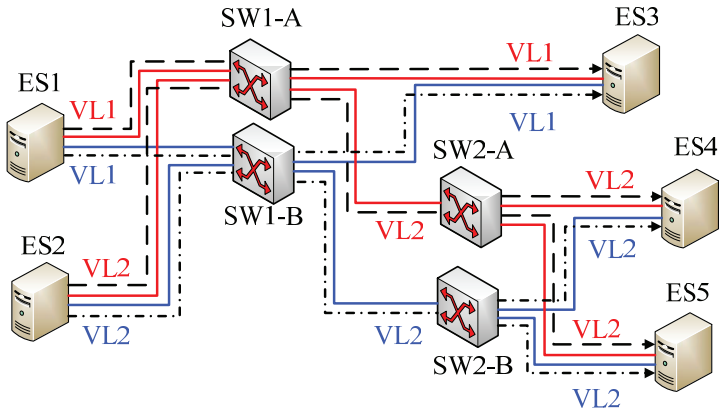


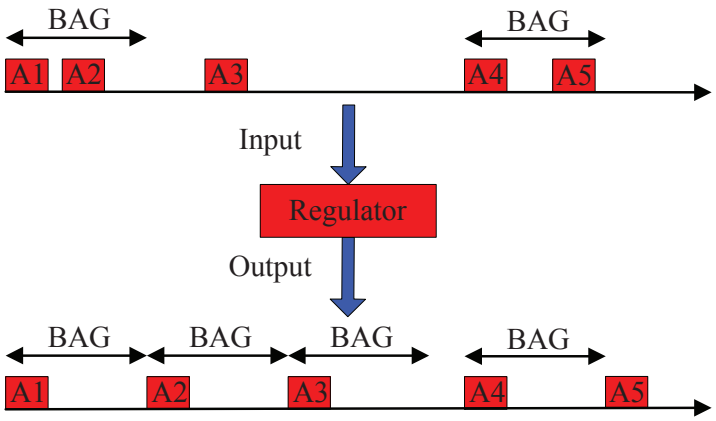
Figure 2.12 An example of AFDX network architecture.

An ES is responsible to generate/receive frames in the AFDX networks, which means that it is either a source or a destination of a VL. Another important function of an ES is to provide guaranteed services in order to perform secure and reliable data exchange with avionic applications as it provides the access interfaces of AFDX networks. As shown in Figure 2.12, each ES is connected to the switches via redundant physical links, denoted by Network A and Network B. A cascaded star topology is applied in switch connections, which makes the network scalable. Usually, it is supposed that the switch has the capability of handling parallel processing. Hence, the packets forwarded to different output ports in a switch do not interfere.

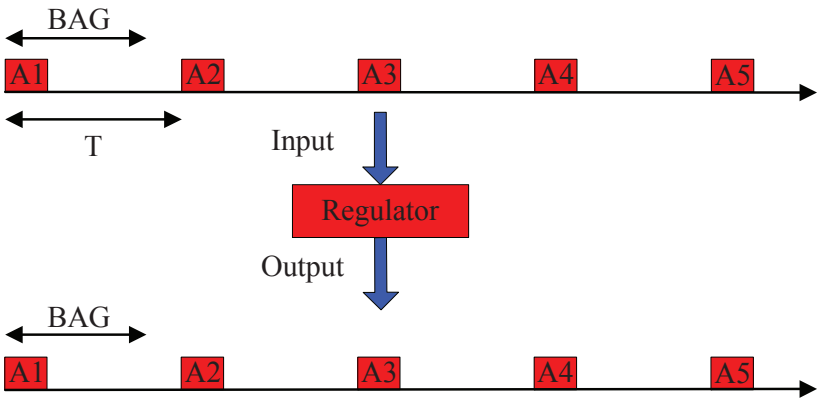
2.6.2 The Concept of VL

A key concept that helps make AFDX deterministic is the Virtual Link (VL), which defines a logical unidirectional connection from one source ES to one or more destination ESs. Note that in AFDX networks, only one ES can be the source of a VL. Every VL is labeled by a predefined unique 16-bit identifier, ranging from 0 to 65535. Besides, in order to provide a consistent performance guarantee for VLs, the routing is statically defined offline. The switch will strictly check the content of each VL's identifier, which indicates its destination address. Only the valid frames are forwarded by the switch to the selected output ports. Furthermore,

the maximum bandwidth allocated to a VL is reserved by its maximum frame size (MFS) and the so-called Bandwidth Allocation Gap (BAG). According to the ARINC 664 standard, the MFS should be in the range of 64 to 1518 bytes. The BAG is the minimum time interval between successive frames in a VL (measured at start time) and should be a power of 2 multiplied by 1 ms within the set {1, 2, 4, 8, 16, 32, 64, 128}(ms). In order to guarantee that the BAG for each VL is respected in source ESs, the mechanism of traffic shaping or regulation is employed as shown in Figure 2.13. The regulator, which aims at limiting the instantaneous frame rate of a VL, controls the data flow of the VL to be delivered with respect to the BAG. Thus the frame input, either periodic or aperiodic, is regulated according to the predefined configuration.



(a) Aperiodic flow regulation



(b) Periodic flow regulation

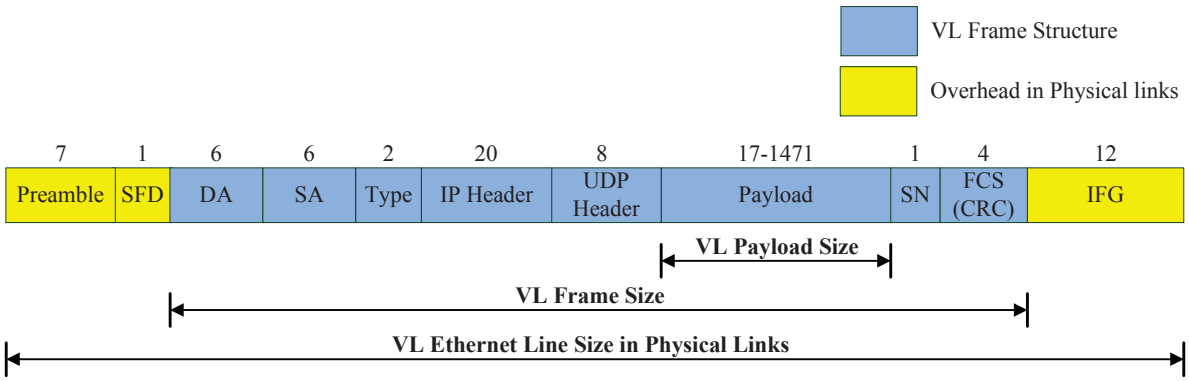
Figure 2.13 The regulation of VL flow.

Note that the VL is allowed to remain silent when there is no data to transmit, no matter

whether the network is busy or not. In addition, the BAG is not necessarily the period of a VL.

2.6.3 The VL Frame Structure

The messages within AFDX networks are transmitted through VLs. As shown in Figure 2.14, the frame of each VL is composed of 6-byte MAC DA, 6-byte MAC SA, 2-byte type field, 20-byte IP header, 8-byte UDP header, an AFDX payload ranging from 17 to 1471 bytes, 1-byte sequence number (SN), and 4-byte frame check sequence (FCS).



Notes: SFD-Start Frame Delimiter, DA-Destination Address, SA-Source Address, SN-Sequence Number, FCS-Frame Check Sequence, IFG-Inter Frame Gap.

Figure 2.14 AFDX frame structure.

The MAC DA is used for routing the frames within switches. The switches and the destination ES(s) accept only the frames associated with DAes in the predetermined configuration table, as the VL identifier is located in the MAC DA. The MAC SA should be compliant with IEEE 802.3 and it indicates to which redundant AFDX network (Network A or Network B) the MAC is connected. In current AFDX standard, IPv4 is employed. Thus, a constant value of 0x0800 is assigned to the type field. Following the type field, IP header is specified to indicate the information, e.g., unicast or multicast transmission. Although the IP header should be compliant with the IPv4 format, the total length for AFDX frame should not include the SN. Therefore, the total length indicated in IP header ranges from 21 to 1499 bytes rather than from 21 to 1500 bytes as in the standard Ethernet. If the payload is less than 17 bytes, padding bytes will be attached. In AFDX frame structure, an SN parameter is added and employed for the integrity checking (IC) and redundancy management (RM). The FCS (or CRC) field is checked in switches and the destination ES(s) to verify the frame validity. During the transmission over physical links, 20 more bytes, including 7 bytes preamble, 1 byte SFD, and 12 bytes IFG, are added on the frames, which need to be considered when

performing the calculation.

2.6.4 Sub-VL Aggregation

According to the ARINC 664-part 7 standard, a VL can be composed of one or up to four Sub-VLs as shown in Figure 2.15. One of the main objectives of Sub-VL aggregation is to improve bandwidth utilization efficiency. Each Sub-VL has a dedicated First-In, First-Out (FIFO) queue. The Sub-VL FIFO queues are read out on a round-robin (RR) basis, by the VL FIFO queue [13]. After aggregation, the frames are regulated according to the BAG of the VL and then delivered.

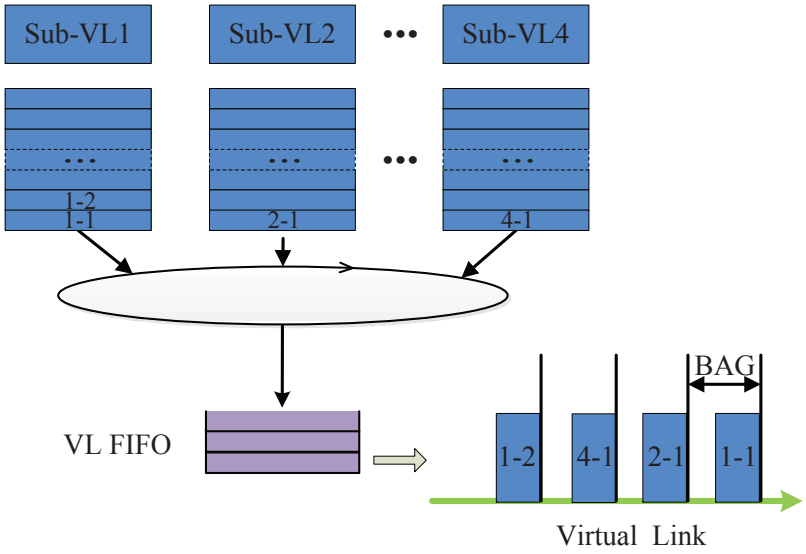


Figure 2.15 Sub-VL aggregation mechanism.

To illustrate how Sub-VL aggregation may optimize bandwidth utilization, we consider the following example in which different types of source data are encapsulated in VLs for transmission. The processing capacity of the source ES is determined by the bandwidth reserved for VLs, which is parameterized by the BAG and the MFS. Suppose for instance that each Sub-VL has a period T=15ms and a MFS $L_{\max}$=1518 Bytes. The simplest configuration is to take every Sub-VL as a VL. Then the VL has the same MFS as the Sub-VL. According to the standard, the BAG should be a power of 2 multiplied by 1ms and selected from the set {1ms, 2ms, 4ms, 8ms, 16ms, 32ms, 64ms, 128ms}. In addition, since no frame should be lost due to buffer overflow, the BAG should be smaller than or equal to T. Thus in the example, to accommodate a source flow of period T=15ms, the BAG of the VL should be 8ms. In Ethernet transmission, an overhead of 20 bytes (IFG+Preamble+SFD) should be added into the size of

VLs. Then the reserved bandwidth for each VL is equal to $(L_{\max}+20)\times 8/\text{BAG}$ =1.538Mbps. Suppose that the physical link operates at 100Mbps. Without considering the jitter at the output, the source ES can transmit at most $\lfloor 100/1.538 \rfloor$=65 VLs. This means that it can manage up to 65 Sub-VLs with a period T=15ms. However, the real bandwidth utilization is $(L_{\max}+20)\times 8/T$=0.82Mbps. Hence, nearly 50 percent bandwidth for every VL is wasted in this example. During transmission, the VLs are frequently in the idle state. Consequently, if more source data is added without aggregation, another source ES is required for this configuration. Instead, if we aggregate three Sub-VLs into one VL, the MFS of the VL does not change. If Sub-VLs with suitable data rate are available, the BAG for an aggregated VL can become 4ms (the detailed computation method can be found in Chapter 4). For each VL, the reserved bandwidth becomes $(L_{\max}+20)\times 8/\text{BAG}$=3.076Mbps. In this configuration, one source ES can manage at most $\lfloor 100/3.076 \rfloor$=32 VLs aggregating in total 96 Sub-VLs. Therefore, without any additional hardware, the processing capability of the source ES can be improved by around 48%, leading to a better bandwidth utilization.

2.6.5 VL Scheduling in Source ES

As source ES supports multiple VLs simultaneously, a scheduler is needed to multiplex different flows coming from the regulators as shown in Figure 2.16. The unregulated flow is either from UDP/IP layer with or without packet fragmentation or from Sub-VL aggregate.

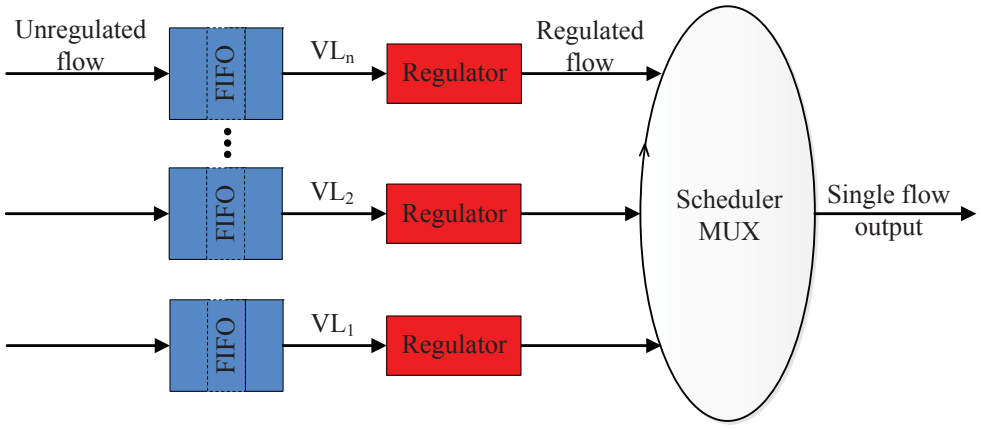


Figure 2.16 Model of VL scheduling.

Since the VLs are independent with each other, jitter may be introduced due to multiplexing. The AFDX standard does not impose a scheduling algorithm, although FIFO is considered as the default scheme. Nevertheless, the scheduler in source ES must guarantee that the

jitter is bounded by 500 μs in all cases to avoid the impact on the determinism of the whole network. A more detailed constraint is given in [13] as follows:

$$\begin{cases} J_{\max} \leq 40\ \mu s + \dfrac{\sum\limits_{i\in\{\text{set of VLs}\}} (20 \text{ bytes} + L^i_{\max}\ bytes) \times 8\ bits/bytes}{\text{Nbw}\ bits/s}, \\ J_{\max} \leq 500\ \mu s, \end{cases} \tag{2.1}$$

where Nbw is medium bandwidth and $L^i_{\max}$ is the MFS of VL_i.

2.6.6 Integrity Checking and Redundancy Management in Destination ESs

As shown in Figure 2.12, VL1 and VL2 are transmitted through two redundant and independent networks to improve the reliability of frame transit. For switches, there is no need to know the redundancy as they are duplicated. In fact, the RM is performed at the destination ES as shown in Figure 2.17.

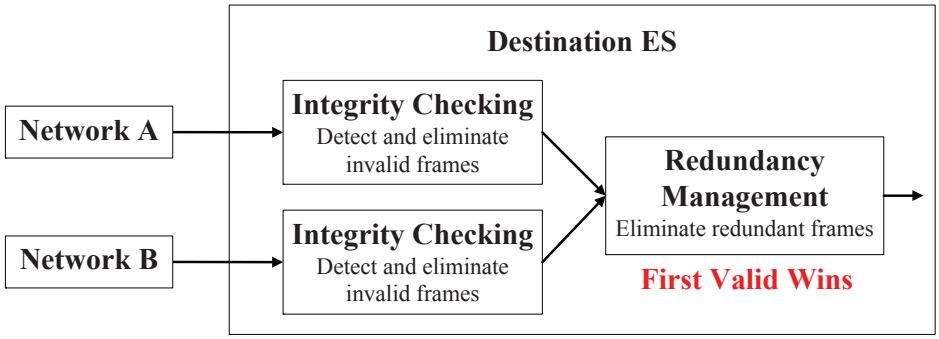


Figure 2.17 Redundancy management in destination ES [13].

The IC is performed before RM to check the sequence number of each frame on its path in the interval: [PSN $\oplus$ 1, PSN $\oplus$ 2], where the previous sequence number (PSN) is the SN of the previously received frame (not necessarily forwarded). The wrap-around operation, $\oplus$, is defined as:

$$\text{PSN} \oplus 1 = (\text{PSN mod } 255) + 1. \tag{2.2}$$

For example, when PSN=254, PSN$\oplus$1=255 and PSN$\oplus$2=1. The IC module guarantees that only the valid frames are forwarded.

In RM, the policy "First Valid Wins" allows the network to tolerate frame loss in either path.

In more detail, two parameters, the sequence number and SkewMax, are used to identify redundant frames. Two frames with identical sequence number are redundant frames, also the arrival time difference between these two frames cannot exceed the value of SkewMax. Otherwise the later reception is identified as a new frame. Hence, SkewMax is the upper bound of transmission delay difference for the redundant frames with identical sequence number.

An example of RM is given in Figure 2.18, in which a frame loss happens on one network, Network B. As shown in the figure, the redundant copies of frames passed through RM module are merely discarded under fault-free reception. In the case of frame loss in either path, the redundancy mechanism associated with the "First Valid Wins" management policy enables the availability of the AFDX networks, which further enhances the reliability of AFDX networks.

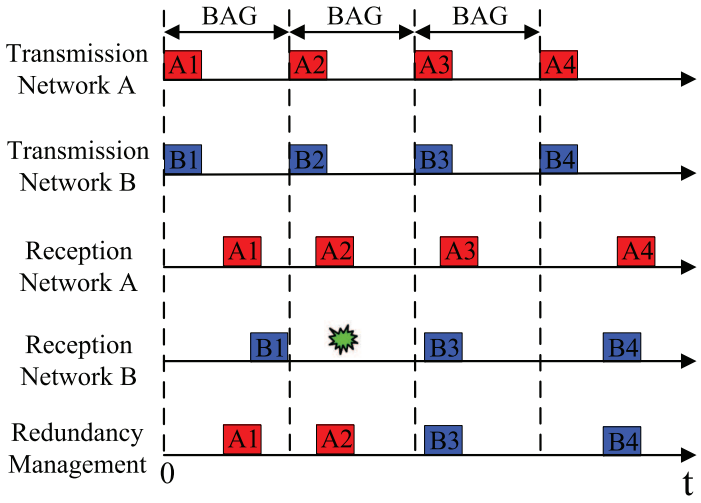


Figure 2.18 An example of RM [13].

2.7 A Comparison of Avionic Network Protocols

In avionic industry, ARINC 429 is the predominant technology for commercial applications and MIL-STD-1553B is the dominant standard in military applications [111]. Both of them have been extended since the initial specifications were published, and consequently they both encounter the same challenges due to the ever increasing performance demands, e.g., transmission speed and bandwidth in modern avionic communication systems. AFDX has been developed initially for commercial aircraft applications. This technology has been adopted as backbone networks in many aircrafts. Another promising technology developed recently for avionic communications is ARINC 825, which is based on CAN 2.0B. It is a cost-effective so-

lution because of the worldwide availability of CAN technology and allows taking advantage of weight savings at the aircraft integration level [14].

Table 2.2 A capability comparison of ARINC 429, MIL-STD-1553B, ARINC 825, TTEthernet, and AFDX [14, 13, 53, 115, 111, 104, 99]

	Topology	Duplex	Maximum Speed	Latency	Maximum Payload	Error Detection
ARINC 429	Bus/Star	Simplex	100 KHz	Fixed	19 bits	Parity Bit
MIL-STD-1553B	Bus/Star	Half-Duplex	1 MHz	Variable	16 bits	Parity Bit
ARINC 825	Bus	Half-Duplex	1 MHz	Bounded	8 Bytes	CRC-15
TTEthernet	Mesh	Full-Duplex	1 GHz	Fixed*	1472 Bytes	CRC-32
AFDX	Mesh	Full-Duplex	100 MHz	Bounded	1471 Bytes	CRC-32

$*$: The latency is fixed for TT traffic.

A comparison on the basic features and characteristics of the aforementioned avionic network technologies is given in Table 2.2. For ARINC 429 and MIL-STD-1553B, the frame size is fixed and the carried payload is limited. Both of them employ a parity bit for error detection, which runs into the risk of concealing some corrupted frames due to its limited capability. In addition, the point-to-point connection of ARINC 429 networks may suffer from design complexity as well as cabling burden when an excessive interconnection is required. Compared with ARINC 429 and MIL-STD-1553B, AFDX provides a higher transmission speed, has the capability to carry more payload, and enables simultaneous bidirectional data transmission. Furthermore, AFDX networks are more flexible as ESs can be easily attached or removed.

Compared with ARINC 825, AFDX offers more payload options ranging from 17-byte to 1471-byte and more bandwidth. The full-duplex mode eliminates the possible collision between transmission and reception and allows the devices to communicate with each other simultaneously. Nevertheless, ARINC 825 is a more convenient technology for field bus due to its cost-effectiveness and ease for development. In fact, AFDX does not intend to replace ARINC 825. In practice, some avionic systems, e.g., in B787, have incorporated ARINC 825 to AFDX networks as an ancillary field bus.

Although AFDX and TTEthernet are both based on IEEE 802.3 technology, they are different variants of the standard Ethernet technology. AFDX is an asynchronous standard, while TTEthernet is a synchronous network, even though AFDX protocol is also supported by TTEthernet according to the standard SAE AS6802. As shown in Table 2.2, the maximum speed of TTEthernet can reach 1 GHz. In fact, AFDX can also support 1 GHz transmission speed [106]. As widely known, TT architectures are based on strong regularity assumptions,

and hence they are less flexible than asynchronous protocols [74]. During TTEthernet design, predefined schedules are required for TT applications to guarantee low transmission jitters. The synthesis of such schedules is known to be an NP-complete problem [105]. This is really challenging in early system design phases, as all the information is not available yet. In contrast, the asynchronism is a feature allowing providing robustness in communications and facilitating design and integration.

The outstanding performance offered by AFDX makes it appealing to serve as the backbone for all avionic systems, including flight controls, cockpit avionics, air-conditioning, power utilities, fuel systems, landing gear, etc. AFDX has been a key avionic communication technology used and considered in many current and future aircrafts, including Airbus A380, A350, A400M, Boeing B787, Sukhoi Superjet 100, ATR 42 & ATR 72 (-600), AgustaWestland AW101, Agusta Westland AW189, Agusta Westland AW169, Irkut MS-21, Bombardier Global Express, Bombardier CSeries, Learjet 85, Comac ARJ21, and AgustaWestland AW149 [2].

CHAPTER 3 TOOLS FOR ANALYSIS AND DESIGN OF DETERMINISTIC AND RELIABLE AVIONIC NETWORKS

In this chapter, the focus is put on introducing tools used in the research presented in this thesis for performance analysis of avionics networks, design optimization, and quantitative reliability assessments of avionic networks.

3.1 Deterministic Network Calculus

The Network Calculus (NC) is a theory of queuing systems that emerged in the 90's and that is now the prominent tool for network performance analysis in time-critical applications. So far, there are two branches for NC: deterministic (or classical) NC and Stochastic Network Calculus (SNC). In this section, we mainly focus on the deterministic NC theory. Some basic concepts introduced in this section are also employed by SNC.

3.1.1 Arrival Curve and Service Curve

In NC, there are two basic concepts to describe the input and output flows in a network node: the arrival curve and the service curve. Their definitions are given below.

Arrival Curve: Let $\alpha(t)$ be a wide-sense increasing function for $t \geq 0$. The flow $R(t)$ is said to be constrained by $\alpha(t)$ if and only if for any $s \leq t$:

$$R(t) - R(s) \leq \alpha(t-s).$$

We say then that $R(t)$ has $\alpha(t)$ as an arrival curve, or $R(t)$ is α-smooth.

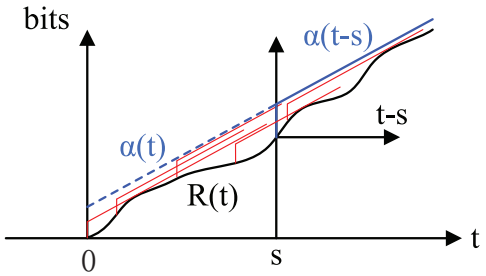


Figure 3.1 Illustration of arrival curve where $R(t)$ is constrained by $\alpha(t)$ in any interval.

As shown in Figure 3.1, $\alpha(t)$ is the upper bound of bit accumulation of $R(t)$ in any time interval. Obviously, there exists a set of curves that meet this condition, which means that

the arrival curve of $R(t)$ is not unique. Therefore in practical analysis, we expect to find the tightest arrival curve to better describe the characteristics of the traffic.

Affine arrival curve (fluid model) and staircase arrival curve are among the most used arrival curves in network performance analysis. An affine arrival curve is defined by:

$$\alpha_A(t) = \begin{cases} \rho t + \sigma, \ t \geq 0 \\ 0, \ \text{otherwise} \end{cases} \tag{3.1}$$

where ρ represents the arrival rate and σ is the burst transmission upper bound. A basic VL data traffic model recommended in the AFDX standard is the affine arrival curve, in which $\sigma = L_{\max} + 20$ and $\rho = \dfrac{\sigma}{\text{BAG}}$. A staircase arrival curve is defined by:

$$\alpha_{T,\tau}(t) = \begin{cases} k \left\lfloor \dfrac{t+\tau}{T} \right\rfloor, \ t, \tau \geq 0 \\ 0, \ \text{otherwise} \end{cases} \tag{3.2}$$

where T denotes the time interval between continuous frames and k is the frame size. Obviously, a staircase arrival curve is also suitable to describe the characteristics of a VL. In more detail, the two parameters can be assigned as $k = L_{\max} + 20 = \sigma$ and $T = \tau = \text{BAG}$. Therefore

$$\alpha_{T,\tau}(t) = \begin{cases} \left\lfloor \dfrac{t + \text{BAG}}{\text{BAG}} \right\rfloor \sigma, \ t \geq 0 \\ 0, \ \text{otherwise.} \end{cases}$$

The two arrival curves corresponding to the same VL configuration are given in Figure 3.2. Although the affine arrival curve has a lot of good properties, e.g., easy to compute, the lack of packet view leads to pessimistic results [25]. In contrast, the staircase arrival curve can reveal the frame regulation and offer tighter performance upper bounds. A more detailed analysis in the context of AFDX with the staircase arrival curve model is given in Chapter 6.

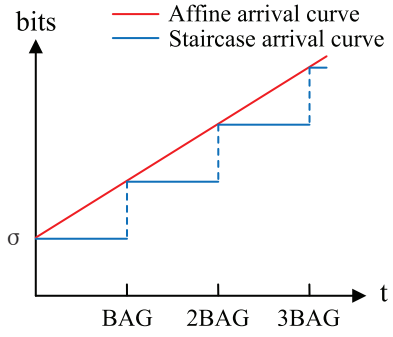


Figure 3.2 Examples of an affine arrival curve and a stair functions arrival curve for a VL.

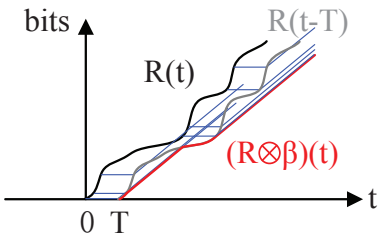
Service Curve: Let $\beta(t)$ be a wide-sense increasing function for $t \geq 0$ with $\beta(0) = 0$. Suppose that a flow through a system has input and output functions $R(t)$ and $R^*(t)$ respectively. Then $\beta(t)$ is the service curve offered by the system if and only if for all $s \leq t$:

$$R^*(t) \geq \inf_{0 \leq s \leq t} (R(s) + \beta(t-s)).$$

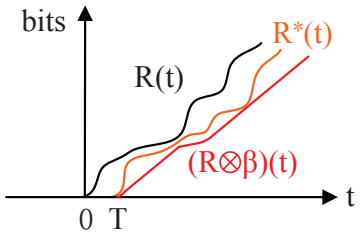
This can also be written as

$$R^*(t) \geq (R \otimes \beta)(t),$$

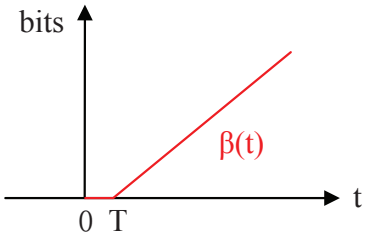
where $\otimes$ denotes the min-plus convolution operation. More details about the min-plus convolution will be introduced in the following subsection. As shown in Figure 3.3, a rate-latency model, which is a dominant model in practical analysis, is considered as a service curve. Suppose that $\beta(t) = C \times [t-T]^+$, where $t \geq 0$ and $[\cdot]^+$ is defined by $\max(\cdot, 0)$. In this case, the rate-latency model has a service rate C and a latency T. Details on how $(R \otimes \beta)(t)$ is computed are illustrated in Figure 3.3(a).



(a) The computation of $(R \otimes \beta)(t)$



(b) $R^*(t)$ is between $R(t)$ and $(R \otimes \beta)(t)$



(c) The model of service curve

Figure 3.3 Illustration of service curve: the output $R^*(t)$ must be lower-bounded by $(R \otimes \beta)(t)$.

Based on the concept of service curve, we are able to describe the service guarantee offered by network elements, e.g., schedulers, during any time interval. Then useful deterministic bounds can be obtained by combining a service curve guarantee with an arrival curve constraint.

3.1.2 Min-Plus Algebra and Basic Performance Bounds

Min-plus algebra is the basis of network calculus. Compared with conventional algebra, min-plus operations treat addition as a computation of the minimum and the multiplication as an addition operation. In the following, two basic operations, namely convolution and deconvolution, are defined and three essential performance bounds are given based on min-plus operation [78].

Min-Plus Convolution: Let $f(t)$ and $g(t)$ be two functions and $f(t) = g(t) = 0$ when $t < 0$. Then the min-plus convolution is defined as:

$$(f \otimes g)(t) = \inf_{0 \le s \le t} \{f(t-s) + g(s)\}.$$

The convolution operation is essential for the analysis of entire networks, because the end-to-end service curve for single traffic passing through cascade systems can be generated by the convolution of individual service curves for each sub-system.

As shown in Figure 3.4, VL_i traverses three cascaded systems to reach its destination. Suppose that for VL_i, the system j offers a service curve $\beta_{Sj}(t) = R_j(t - T_j)^+$ $(j = 1, 2, 3)$, where R_j is the service rate and T_j is the latency. Then, the end-to-end service curve can be expressed as

$$\beta^i_{e2e} = \beta_{S1} \otimes \beta_{S2} \otimes \beta_{S3}.$$

First, we perform the computation of $\beta_{S1} \otimes \beta_{S2}$ as follows:

$$\begin{aligned}
\beta_{S1} \otimes \beta_{S2} &= \inf_{0 \le s \le t} \{\beta_{S1}(t-s) + \beta_{S2}(s)\} \\
&= \inf_{0 \le s \le t} \left\{R_1(t - s - T_1)^+ + R_2(s - T_2)^+\right\} \\
&= \min \left\{R_1(t - T_2 - T_1)^+, R_2(t - T_1 - T_2)^+\right\} \\
&= \min \{R_1, R_2\} \times (t - T_2 - T_1)^+.
\end{aligned}$$

Similarly, we have

$$\beta^i_{e2e} = \min\{R_1, R_2, R_3\} \times (t - T_1 - T_2 - T_3)^+.$$

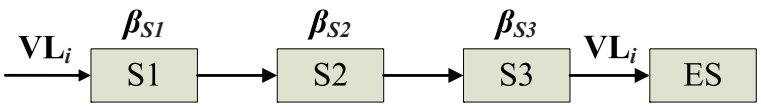


Figure 3.4 An example of service curve deduction using convolution in cascade systems.

Obviously, it is easier to compute the end-to-end delay bound based on the concatenated service curve and the result is tighter than the summation of the delay at each sub-system since the burst is taken into account only once, which is known as the principle of "Pay Bursts Only Once" (PBOO). Detailed explanation about the principle of PBOO can be found in [78].

Min-Plus Deconvolution: Let $f(t)$ and $g(t)$ be two functions and $f(t) = g(t) = 0$ for all $t < 0$. Then the min-plus deconvolution is defined as:

$$(f \oslash g)(t) = \sup_{u \geq 0}\{f(t+u) - g(u)\}.$$

With the min-plus deconvolution, three important bounds in network calculus can be easily expressed.

Output Flow Bound: Suppose that a flow constrained by an arrival curve $\alpha(t)$ traverses a system, which offers a service curve of $\beta(t)$. Then, its output flow is bounded by the arrival curve $\alpha^*(t) = (\alpha \oslash \beta)(t)$, $t \geq 0$. Detailed proof can be found in [78].

For example, a VL constrained by $\alpha(t) = \rho t + \sigma$, $t \geq 0$, traverse a system, which offers a service curve $\beta(t) = C(t-\tau)^+$, $\tau, t \geq 0$ and $C > \rho$. Then the deconvolution of the arrival curve and the service curve can be given as follows:

$$\begin{aligned}(\alpha \oslash \beta)(t) &= \sup_{u \geq 0}\{\alpha(t+u) - \beta(u)\} \\ &= \sup_{u \geq 0}\{\alpha(t+u) - C(u-\tau)^+\} \\ &= \max\left\{\sup_{0 \leq u \leq \tau}\{\alpha(t+u)\}, \sup_{u > \tau}\{\alpha(t+u) - Cu + C\tau)\}\right\}.\end{aligned} \tag{3.3}$$

When $t \leq -\tau, \alpha(t+\tau) = 0$. Thus, (3.3) becomes:

$$\begin{aligned}(\alpha \oslash \beta)(t) &= \max\left\{0, \sup_{\tau < u \leq -t}\{\alpha(t+u) - Cu + C\tau\}, \sup_{u > -t}\{\alpha(t+u) - Cu + C\tau\}\right\} \\ &= \max\{0, 0, \sigma + Ct + C\tau\} \\ &= (\sigma + Ct + C\tau)^+.\end{aligned} \tag{3.4}$$

When $t > -\tau$, (3.3) becomes:

$$\begin{aligned}(\alpha \oslash \beta)(t) &= \max\left\{\alpha(t+\tau), \sup_{u > \tau}\{\alpha(t+u) - Cu + C\tau)\}\right\} \\ &= \alpha(t+\tau).\end{aligned} \tag{3.5}$$

The deconvolution results are shown in Figure 3.5. According to the definition of arrival

curve, the output flow to be fed into the next system is constrained by:

$$(\alpha \oslash \beta)(t) = \alpha(t+\tau),\ t \geq 0. \tag{3.6}$$

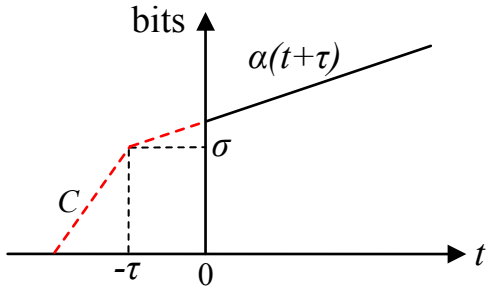


Figure 3.5 Deconvolution results and the arrival curve for the output flow.

Backlog Bound: Suppose that a flow constrained by an arrival curve $\alpha(t)$ traverses a system, which offers a service curve of $\beta(t)$. Then, the backlog $B(t)$ for all t is constrained by $v(\alpha, \beta) = \sup_{u \geq 0}\{\alpha(u) - \beta(u)\} = (\alpha \oslash \beta)(0)$.

In other words, the maximum backlog equals to the maximum vertical deviation between $\alpha(t)$ and $\beta(t)$.

Delay Bound: Suppose that a flow constrained by an arrival curve $\alpha(t)$ traverses a system, which offers a service curve of $\beta(t)$. Then, the virtual delay $D(t)$ is constrained by $h(\alpha, \beta)$ given by

$$h(\alpha, \beta) = \inf\{d | (\alpha \oslash \beta)(-d) \leq 0\}.$$

As shown in Figure 3.5, $d_{\max} = \sigma/C + \tau$ is the value verifying $(\alpha \oslash \beta)(-d_{\max}) = 0$.

The concepts and performance bounds introduced in this section will be employed in the subsequent chapters analyzing the performance of AFDX networks.

3.2 Stochastic Network Calculus

Deterministic NC provides safe upper bounds for safety-critical applications under the worst-case scenarios. However, the obtained delay bounds are pessimistic in most of the practical cases. Hence, the overestimation of delay upper bounds leads to inefficient utilization of network resources. Unlike deterministic analysis, SNC takes into account the stochastic nature of traffic and service processes in the analysis to better make use of statistical multiplexing gains [68]. It provides probabilistic delay bounds with certain violation probabilities, which

are typically tighter than the upper bounds obtained by deterministic NC. Thus, an immediate benefit of probabilistic analysis is that it allows relaxing the constraints in schedulability assessment and consequently help increase the network utility.

In general, the stochastic performance metric can be expressed as

$$\Pr\{\text{Performance is worse than a certain bound}\} \leq \varepsilon.$$

For example, $\Pr\{Q(t) > b\} \leq \varepsilon$, where $Q(t)$ is the backlog at time t and b is a constant. In fact, the deterministic NC can be interpreted as a special case of SNC, in which $\varepsilon = 0$. In order to obtain probabilistic performance bounds, normally a probabilistic arrival curve or service curve is required. For some special applications, e.g., AFDX networks, performance bounds can also be deduced based on the application of some inequalities. In the following, we introduce an approach to obtain stochastic bounds on backlog and delay based on Hoeffding's inequalities.

A framework to derive probabilistic guarantees for aggregate flows in networks has been developed in [124] and [125], which can be applied to AFDX networks to provide more realistic and tighter delay bounds compared to the deterministic ones.

Consider a set of VLs sharing the same node denoted by $\mathcal{I} = \{1, 2, ..., I\}$. Suppose that $A_i(t)$ and $A_i^*(t)$, for $i \in \mathcal{I}$, are the accumulation bit numbers of VL_i at time t on the input and output, respectively. Then $A(t) = \sum_{i=1}^{I} A_i(t)$ can be interpreted as the bit aggregation of all VLs on the input. Likewise, define $A^*(t) = \sum_{i=1}^{I} A_i^*(t)$ as the bit aggregation of all VLs at the output. Define $[g(t)]^+ = \max\{g(t), 0\}$. We make the following assumptions:

(A1) A_i and A_j are independent, $\forall i, j \in \mathcal{I}$ and $i \neq j$.

(A2) For $A_i(t)$, $i \in \mathcal{I}$, there exists an arrival curve $\alpha_i(t)$, such that:

$$A_i(t) - A_i(s) \leq \alpha_i(t - s), \forall s, t \in \mathbb{R}, \tag{3.7}$$

where $\alpha_i(t) = 0$, $\forall\ t < 0$. Then $\alpha(t) = \sum_{i=0}^{I} \alpha_i(t)$ is the arrival curve of $A(t)$. Let $\alpha_i(t) = \rho_i t + \sigma_i$, where σ_i is the maximum frame length $L_{\max_i}$ of VL_i and $\rho_i = L_{\max_i}/\text{BAG}_i$. BAG_i is the minimum time interval between two consecutive frames of VL_i. Then $\alpha(t)$ is given by $\alpha(t) = \rho t + \sigma$, where $\rho = \sum_{i=1}^{I} \rho_i$ and $\sigma = \sum_{i=1}^{I} \sigma_i$.

(A3) The node offers a service curve $\beta(t) = R(t - T)^+$ for $A(t)$, where T is the worst-case latency, and R is the physical link transmission rate.

(A4) There exists $\tau < \infty$ and $\tau = \inf\{t \geq 0 | \alpha(t) \leq \beta(t)\}$. τ is the intersection of the arrival curve α and service curve β.

Let $Q(t)$ be the backlog of $A(t)$ at time t:

$$Q(t) = \sup_{-\infty \leq s \leq t} \{A(t) - A(s) - \beta(t-s)\}. \tag{3.8}$$

$Q(0)$ is then the backlog incurred by an arbitrary frame that arrives at time 0. Let $d(0)$ be the delay encountered by an arbitrary frame of the VLs that arrives at time 0.

For homogeneous traffics, we have $\alpha_i(t) = \alpha_j(t)$, $\forall i, j \in \mathcal{I}$. Under the assumption of (A1)-(A4), the probability that $Q(0)$ exceeds a given number b can be bounded by:

$$\Pr(Q(0) > b) \leq \sum_{k=0}^{K-1} \exp(-I \times g(s_k, s_{k+1})), \tag{3.9}$$

where $K \in \mathbb{N}$, and $0 = s_0 \leq s_1 \leq \cdots \leq s_K = \tau$. The function $g(u, v)$ is defined as:

$$g(u,v) = \begin{cases} +\infty, \ b > \alpha(v) - \beta(u); \\ 0, \qquad b < \rho v - \beta(u); \\ \dfrac{\beta(u)+b}{\alpha(v)} \ln \dfrac{\beta(u)+b}{\rho v} + \left(1 - \dfrac{\beta(u)+b}{\alpha(v)}\right) \ln \dfrac{\alpha(v)-\beta(u)-b}{\alpha(v)-\rho v}, \ \text{otherwise.} \end{cases} \tag{3.10}$$

Under the assumption of (A1)-(A4) for heterogeneous traffics, we have:

$$\Pr(Q(0) > b) \leq \sum_{k=0}^{K-1} \exp(-g(s_k, s_{k+1})). \tag{3.11}$$

where $K \in \mathbb{N}$, and $0 = s_0 \leq s_1 \leq \cdots \leq s_K = \tau$. The function $g(u, v)$ is defined by:

$$g(u,v) = \frac{2\left([q + \beta(u) - \rho v]^+\right)^2}{\sum_{i=1}^{I} \alpha_i(v)^2}. \tag{3.12}$$

The distribution of the backlog that exceeds a given value at the arrival time of a frame, namely Palm probability, is denoted by $\Pr_A$. According to [124], we have:

$$\Pr_A(Q(0) > b) \leq \frac{R}{\rho} \Pr(Q(0) > b). \tag{3.13}$$

Under the assumptions of A(1)-A(4), if a frame arrives at time 0, the upper bound of the delay probability for both homogeneous and heterogeneous traffics can be expressed as:

$$\Pr(d(0) > t) \leq \Pr_A(Q(0) > Rt) \leq \frac{R}{\rho}\Pr(Q(0) > Rt). \tag{3.14}$$

3.3 Approaches for Multi-objective Optimization

In network design, optimization problems are often characterized by multiple objectives. In this section, the multi-objective optimization methods are briefly introduced and approaches for achieving optimal solutions are presented.

3.3.1 Multi-objective Optimization Problem

Consider a problem with m objective functions $f_i(x)$, $i = 1, \ldots, m$, where $x \in \mathbb{R}^n$ is a vector of decision variables. Therefore, a multi-objective optimization problem can be formulated as [36]:

$$\begin{aligned} &\min_x F(x) = [\, f_1(x), f_2(x), \ldots, f_i(x), \ldots, f_m(x)) \,]^T \\ &\text{s.t.} : x \in X, \end{aligned} \tag{3.15}$$

where $X \subset \mathbb{R}^n$ is the feasible design space. Typically, the feasible design space X is defined by a number of inequality and equality constraints, e.g., $X = \{x \in \mathbb{R}^n \ | g_j(x) \leq 0,\ j = 1, 2, \ldots, n_i;\ h_k(x) = 0,\ k = 1, 2, \ldots, n_e\}$. This means that every feasible vector x must be compliant with the constraints. Furthermore, the feasible criterion space is defined as $Z = \{F(x) | x \in X\}$.

3.3.2 Pareto Optimality for Multi-objective Problems

Typically in multi-objective optimization, there is no single global solution that minimizes all objective functions simultaneously, because the optimization problem may have competing or conflicting objectives. In this case a gain in one objective may lead to degrading other objectives. Consequently, an optimal solution to this problem can only be achieved in the sense of Pareto optimality. Specifically, there are two variations of Pareto optimality: Pareto optimality and weak Pareto optimality.

Pareto optimality[91]: A point $x^* \in X$ is called a Pareto optimal solution if and only if there is no other point $x \in X$ such that $F(x) \leq F(x^*)$, and $f_i(x) < f_i(x^*)$ for at least one function.

Obviously, all the optimal solutions locate in the feasible criterion space Z and the set of all Pareto optimal solutions is defined as Pareto front. According to the definition of Pareto optimality, it is clear that the Pareto front lies on the boundary of Z. An example of Pareto front with two objective functions is illustrated in Figure 3.6.

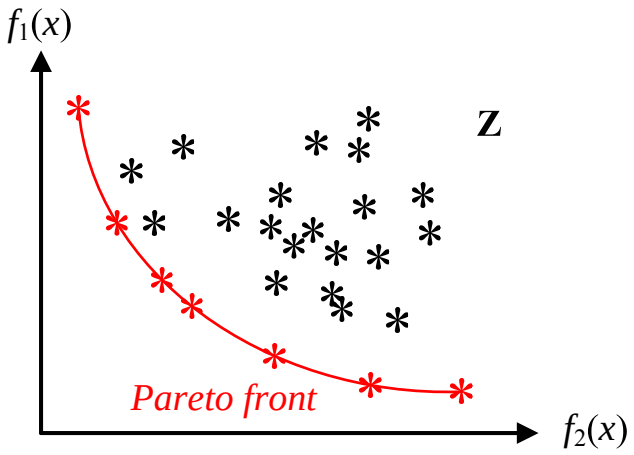


Figure 3.6 An example of Pareto front.

Weak Pareto optimality[91]: A point $x^* \in X$ is called a weak Pareto optimal solution if and only if there is no other point $x \in X$ such that $F(x) < F(x^*)$.

A Pareto optimal point implies that there does not exist another point that can improve any objective function without degrading at least one of the other objectives, while a weak Pareto optimal point implies that there does not exist another point that improves all the objective functions simultaneously. Therefore, all Pareto optima are weak Pareto optima. Reversely, weak Pareto optimal points are not necessarily Pareto optimal.

3.3.3 Lexicographic Method

In order to find solutions in the sense of Pareto optimality, it is necessary to impose design preferences to reflect the relative importance of different objectives. To this end, different methods, such as the weighted sum, the weighted min-max, and the lexicographic process, have been proposed, which allow the system designers to specify preferences. Although the methods with weight coefficients are widely applied, there is no available formal theoretical analysis on how to chose these parameters [136]. The weight coefficients are normally determined based on engineering experiences, simulation results, or experiments, which is inflexible and complicated. Another popular method, the lexicographic process, is a priority-driven framework, in which all the objective functions are sorted in the order of their relative

importance. Compared to the weighted sum method or the weighted min-max method, the lexicographic method is more suitable for multi-objective optimization problems considered in this thesis.

In general, the lexicographic process can be formulated as follows:

$$\begin{aligned} &\min_{x \in X} f_i(x); \\ &\text{s.t.} : f_j(x) \leq f_j(x_j^*), j = 1, 2, \ldots, i-1, i > 1; \\ &\qquad i = 1, 2, \ldots, m; \end{aligned} \tag{3.16}$$

where i represents the sequence of the objectives, $f_j(x_j^*)$ is the optimum of the jth objective function, and m is the number of the objectives. Note that $f_j(x_j^*)$ is not necessarily constant as new constraints are continuously introduced with the increase of i.

A variation of the lexicographic approach is to introduce a parameter, δ, which aims at relaxing the constraints. The corresponding formulation is given by [126, 107]:

$$\begin{aligned} &\min_{x \in X} f_i(x); \\ &\text{s.t.} : f_j(x) \leq (1 + \delta_j) f_j(x_j^*), j = 1, 2, \ldots, i-1, i > 1; \\ &\qquad i = 1, 2, \ldots, m; \end{aligned} \tag{3.17}$$

where $\delta_j \geq 0$. Then, δ_j is imposed to control the trade-offs among the objective functions. Different Pareto optimal solutions may be found by changing the value of δ_j. Consequently, the δ-constraint approach is less sensitive to the initial ranking of the objective functions, which may lead to better solutions.

3.4 Fault Tree Analysis-based Reliability Assessment

Fault Tree Analysis (FTA) is one of the most prominent top-down analysis techniques for reliability assessment, in which a qualitative model for undesired events is constructed and then investigated [15, 37]. This technique focuses mainly on determining the origin of failures and their propagation. Typically, a fault tree is composed of event symbols and logic symbols.

3.4.1 Event Symbols

The most commonly used symbols for events are shown in Figure 3.7, which include circle, house, diamond, oval, rectangle, and triangle representing respectively various events called: basic, external, undeveloped, conditional, intermediate, and transfer in/out [15, 37].

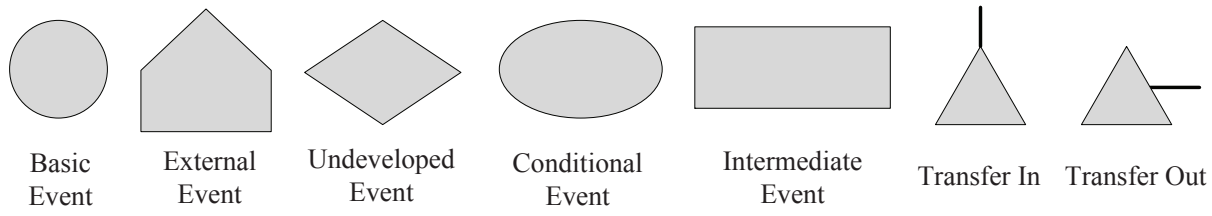


Figure 3.7 Fault tree event symbols.

In general, the symbols in Figure 3.7 are classified into three types: primary event symbols, intermediate event symbols, and transfer symbols.

Circle, house, diamond, and oval are all the primary event symbols, and each of them is associated with a probability for computing the probability of the occurrence of the top event. A circle represents a basic event without further development. A house signifies an external event, which is normally expected to occur. A diamond indicates an undeveloped event, which is not further developed as the necessary details are unavailable. An oval represents a conditional event, which defines specific conditions for a failure mode to occur.

Intermediate events are represented by rectangles, which are used for the description or the documentation of logic symbol outputs or events.

The triangle includes two different types: "transfer in" and "transfer out". A "transfer in" symbol indicates that the fault tree is further developed at the corresponding "transfer out". A "transfer out" symbol signifies that this part of fault tree should be tied to the corresponding "transfer in".

3.4.2 Logic Symbols and Basic Mathematical Operations with Probabilities

The two most common logic gates used to describe the relationship of failure effects are the AND-gate and the OR-gate as shown in Figure 3.8 [15, 37].

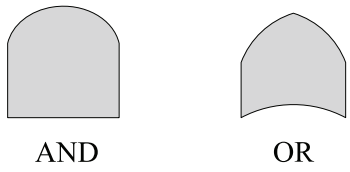
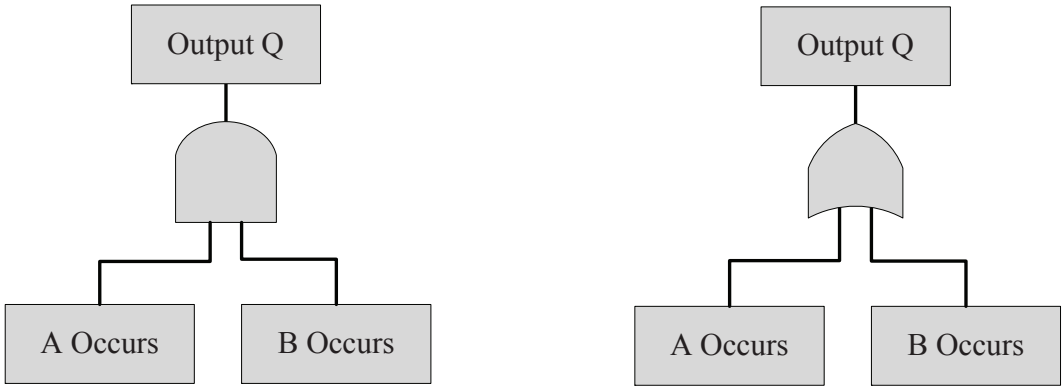


Figure 3.8 Fault tree logic symbols.

An AND-gate is used when the output fault can only occur if all of the input faults take



(a) AND relationship (b) OR relationship

Figure 3.9 An illustration of logic relationships.

place simultaneously. An OR-gate is applied when the output occurs if any one or more of the input faults occur. An illustration of logic relationships is given in Figure 3.9.

Suppose that the failures of the components, A and B, are two mutually independent events. Then for the AND relationship shown in Figure 3.9a, the output probability can be computed as

$$\Pr(\mathrm{Q}) = \Pr(\mathrm{A})\Pr(\mathrm{B}). \tag{3.18}$$

This formula can be extended to more events if they are mutually independent. However, the events encountered in practice may not be mutually independent but only interdependent. In this scenario, the conditional probability ($\Pr(\mathrm{A}|\mathrm{B})$ or $\Pr(\mathrm{B}|\mathrm{A})$) is introduced, which represents the occurrence probability of one event when the other occurs. Therefore, the probability of the output Q can be expressed as:

$$\Pr(\mathrm{Q}) = \Pr(\mathrm{A})\Pr(\mathrm{B}|\mathrm{A}) = \Pr(\mathrm{B})\Pr(\mathrm{A}|\mathrm{B}). \tag{3.19}$$

If A and B are mutually independent, then $\Pr(\mathrm{B}|\mathrm{A}) = \Pr(\mathrm{B})$ and $\Pr(\mathrm{A}|\mathrm{B}) = \Pr(\mathrm{A})$. Consequently, (3.19) reduces to (3.18).

Consider an OR relationship as shown in Figure 3.9b. If A and B are mutually exclusive, then the probability of output Q can be given as

$$\Pr(\mathrm{Q}) = \Pr(\mathrm{A}) + \Pr(\mathrm{B}). \tag{3.20}$$

This equation can be extended to any number of mutually exclusive events with OR relationship. For more general scenarios in which the events are not mutually exclusive, the

expression of Q becomes

$$\Pr(\text{Q}) = \Pr(\text{A}) + \Pr(\text{B}) - \Pr(\text{A and B}). \tag{3.21}$$

If A and B are mutually exclusive, then Pr(A and B) = 0. Consequently, (3.20) reduces to (3.21).

In summary, a fault tree can be constructed by considering the logical relationship of basic events leading to the predefined top event. By establishing a fault tree model, reliability assessment can be carried out in a quantitative manner.

CHAPTER 4 ARTICLE 1: DETERMINISM ENHANCEMENT OF AFDX NETWORKS VIA FRAME INSERTION AND SUB-VIRTUAL LINK AGGREGATION

Determinism is one of the main features of networks applied in real-time safety critical applications. Thus guaranteeing the determinism is especially important in terms of performance, safety and certification. In this chapter, a mechanism based on frame insertion is introduced to improve the determinism of AFDX. The following sections are the reproduction of [83], which has been published in IEEE Transactions on Industrial Informatics.

Authors—Meng Li, Michaël Lauer, Guchuan Zhu, *Senior Member, IEEE,* and Yvon Savaria, *Fellow, IEEE.*

Abstract—AFDX is a standard proposed to implement deterministic networks by providing predictable performance guarantees. The determinism is enforced through the concept of Virtual Link, which defines a logical unidirectional connection between End Systems. Although an upper bounded end-to-end delay can be obtained by using analysis based on, e.g., Network Calculus, frame arrival uncertainty in destination End System is a source of non-determinism that introduces a problem with respect to real-time fault detection. In this paper a mechanism based on frame insertion is proposed to enhance the determinism of frame arrival within AFDX networks. In order to mitigate network load increase due to frame insertion, a Sub-Virtual Link aggregation strategy, formulated as a multi-objective optimization problem, is introduced. In addition, a brute force algorithm, a greedy algorithm, and a greedy algorithm with pre-processing have been developed to find solutions to the optimization problem. Experiments are carried out and the reported results confirm the validity and applicability of the developed approaches.

Index Terms—Avionics Full Duplex Switched Ethernet (AFDX) networks, determinism, optimization, Sub-Virtual Link aggregation.

4.1 Introduction

Determinism, fault tolerance, and timing constraint are the main concerns for critical industrial applications (See, e.g., [16, 64, 29, 60]). Due to stringent performance requirements in safety-critical avionics systems, several new network technologies have been developed these years, among which we can find Avionics Full Duplex Switched Ethernet (AFDX). AFDX has been proposed to meet increasing requirements of high speed, high reliability, and low

cost avionics communication systems. This technology is standardized in ARINC 664 Part 7 [13] and is deployed in many current and future aircrafts such as Airbus A380, A350, A400M, Boeing B787, Comac ARJ21, and Bombardier CS100.

AFDX is a specialization of Ethernet whose purpose is to provide a more deterministic network with predictable performance guarantees. This determinism is enforced mainly through the concept of Virtual Link (VL), inspired by the concept of asynchronous transfer mode (ATM). As stated in the standard, a VL is a conceptual communication link, which defines: (1) a logical unidirectional connection from one source End System (ES) to one or more destination ESs; (2) a maximum bandwidth allocated to this connection. Essentially, two mechanisms are used to ensure that the bounded data transmission bandwidth is respected. At the ingress of the network, i.e. ESs, traffic shaping is used to control the flow for each VL in accordance with the so-called Bandwidth Allocation Gap (BAG), which defines the minimum time interval between successive frames in a VL. In the switches, traffic policing is used to protect the network from babbling-idiot failures. Furthermore, as the routes of the VLs are statically defined off-line, the network offers a consistent performance guarantee. In addition, AFDX is composed of two independent and redundant networks, which provides the high reliability required for ensuring its determinism.

AFDX networks aim at providing a guaranteed service with a firm, mathematically provable, upper bound on end-to-end frame transit delay. Hence the end-to-end delay analysis is considered as a pivotal issue among the mandatory certifications. Much work has been dedicated to evaluate the delay upper bounds. The theoretical methods, including network calculus [40, 41, 78, 24, 109], trajectory approach [19, 21, 20, 65, 22] and response time analysis [119] are applied to the worst-case transmission delay analysis. Scheduling schemes for ESs and switches are proposed to improve the end-to-end delay [66, 135, 67, 82]. Furthermore, simulation and modeling approaches are implemented to evaluate end-to-end delays and to provide experimental upper bounds [33, 134, 26, 6]. With the upper bounded delay, the minimum interval between successive frames in destination ES becomes deterministic.

Nevertheless, there still exist some sources of non-determinism in AFDX networks. First, being an asynchronous protocol, a global time cannot be defined or used throughout the network. Note that the asynchronism is a feature of this network, which has been chosen in order to provide robustness in communications and to facilitate the design of applications using the network. A second source of non-determinism is related to fault detection in the destination ESs. Indeed, the AFDX standard does not force a VL to transmit frames if there is no data to transmit, even though the VL is available. This means that destination ESs cannot detect one or several consecutive frame losses (due to frame corruptions or device

malfunctions on both redundant networks) until a valid frame arrives. For safety-critical applications, this raises a serious issue in terms of determinism and reliability. The motivation of this paper is then to enhance the determinism of AFDX networks by proposing a solution to frame arrival uncertainty.

The proposed solution is based on the idea of inserting filler frames in a VL when its source is silent. This allows destination ESs of the VL to detect a fault if a frame is missing from the periodical pattern obtained with filler frames. Obviously, this mechanism does not affect the maximum bandwidth reserved for a VL and the worst-case performance of a regulated VL. However, inserting filler frames will increase network load and the average bandwidth used by a VL. In order to mitigate the impact on the overall network performance, we leverage a feature described in the AFDX standard, namely Sub-Virtual Link (Sub-VL) aggregation. We show that Sub-VLs aggregation in source ESs allows optimizing the bandwidth utilization of VLs. A Sub-VL aggregation strategy, formulated as a multi-objective optimization problem aimed at minimizing the overhead due to filler frame insertion and the delay introduced by Sub-VL aggregation, is then presented. It is worth noting that the proposed formulation can be applied to the generic Sub-VL aggregation problem in AFDX network design and to the extent of our knowledge, little work is dedicated to the optimization of Sub-VL aggregation. The viability and the applicability of the proposed strategy are demonstrated through numerical simulations. Several algorithms used to reach or approach an optimal solution are developed, including a brute force algorithm (an exhaustive search), a greedy algorithm, and a greedy algorithm with pre-processing. Note that in this paper, the impact on bandwidth due to filler frame insertion and its optimization with Sub-VL aggregation are only considered at the source ES level. The impact on the global network will be studied in future work. It is proposed in [9] to aggregate messages into super-messages in source operating system (OS) partitions defined in the ARINC 653 standard to minimize bandwidth consumption. However, the proposed aggregation is not related to optimizing frame insertion. Moreover, the problem we consider is tackled in a network layer where message aggregation cannot be performed.

The contributions of this paper are:

— a mechanism based on frame insertion that enables real-time fault detection in destination ESs, thus enhancing the determinism of the network;

— a Sub-VL aggregation strategy that mitigates the network load increase due to frame insertion while simultaneously minimizing the delay introduced by Sub-VL aggregation.

The remaining of the paper is organized as follows. Section 4.2 describes issues related to Sub-VL aggregation in AFDX networks and the non-determinism in VL transmission. Sec-

tion 4.3 presents a mechanism for determinism enhancement in AFDX networks. Then, in Section 4.4 the problem of Sub-VL aggregation is formulated and effective algorithms for resolving the corresponding multi-objective optimization problem are developed. In Section 4.5, experimentations are carried out to validate the feasibility of the proposed mechanism and to evaluate the obtained performance. Finally, some concluding remarks and directions for future research are provided in Section 4.6.

4.2 Sub-VL Aggregation and Non-Determinism in VL Transmission

We present in this section the parameter calculation for Sub-VL aggregation in AFDX networks and formulate the delay due to this operation. The non-determinism issue in VL transmission will be discussed, which leads to a suggestion for determinism enhancement.

4.2.1 Sub-VL Aggregation

One of the main objectives of Sub-VL aggregation is to improve the bandwidth utilization efficiency. According to the ARINC 664-part 7 standard, a VL can be composed of one or up to four Sub-VLs. Each Sub-VL has a dedicated First-In, First-Out (FIFO) queue. The Sub-VL FIFO queues are read out on a round-robin (RR) basis, as shown in Figure 4.1, by the VL FIFO queue [13]. After aggregation, the frames are sent according to the BAG of the VL.

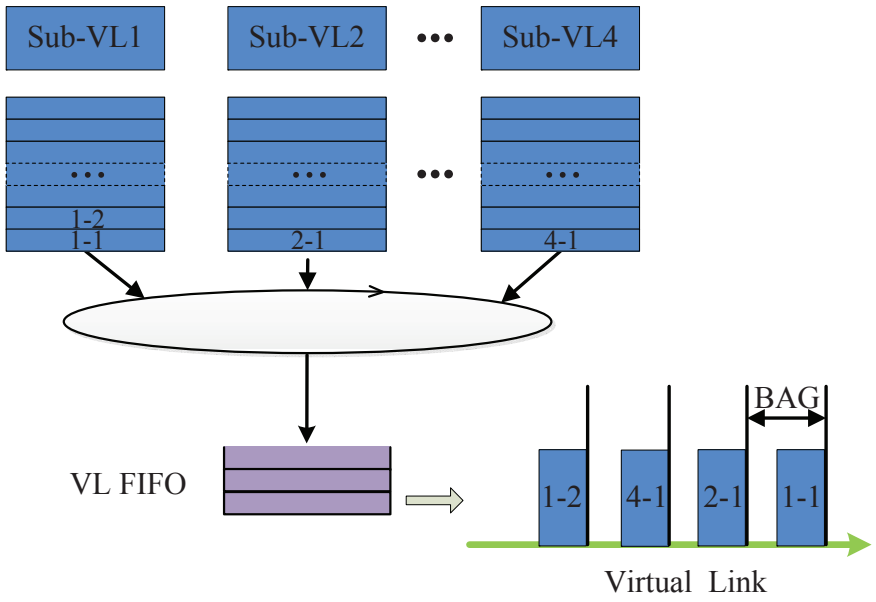


Figure 4.1 Sub-VL aggregation mechanism.

Essentially, a Sub-VL can be dedicated to a source flow from OS partitions, which can be periodic or sporadic. In either case, to allocate bandwidth for each VL, represented by the BAG, the system integrator must set the following two parameters:

— $l^{\max}$: the maximum frame size (MFS) of the source flow;

— T: the minimum time interval between two consecutive frames.

To illustrate how Sub-VL aggregation may optimize bandwidth utilization, we consider the following example in which different types of source data are encapsulated in VLs for transmission. The processing capacity of the source ES is determined by the bandwidth reserved for VLs, which is parameterized by the BAG and the MFS. Suppose for instance that each Sub-VL has a period T=15ms and a MFS $l^{\max}$=1518 Bytes. The simplest configuration is to take every Sub-VL as a VL. Then the VL has the same MFS as the Sub-VL. According to the standard, the BAG should be a power of 2 multiplied by 1ms and selected from the set {1ms, 2ms, 4ms, 8ms, 16ms, 32ms, 64ms, 128ms}. In addition, since no frame should be lost due to buffer overflow, the BAG should be smaller than or equal to T. Thus in our example, to accommodate a source flow of period T=15ms, the BAG of the VL should be 8ms. In Ethernet transmission, an overhead of 20Bytes (Interframe Gap+Preamble+Start Frame Delimiter) should be added into the size of VLs. Then the reserved bandwidth for each VL is equal to $(l^{\max} + 20) \times 8/\text{BAG}$ =1.538Mbps. Suppose that the physical link operates at 100Mbps. Without considering the jitter at the output, the source ES can transmit at most $\lfloor 100/1.538 \rfloor$=65 VLs. This means that it can manage up to 65 Sub-VLs with a period T=15ms. However, the real bandwidth utilization is $(l^{\max} + 20) \times 8/T$=0.82Mbps. Hence, nearly 50 percent bandwidth for every VL is wasted in this example. During transmission, the VLs are frequently in the idle state. Consequently, if more source data are added without aggregation, another source ES is required for this configuration. Instead, if we aggregate three Sub-VLs into one VL, the MFS of the VL does not change. If Sub-VLs with suitable data rate are available, the BAG for an aggregated VL can become 4ms (this can be shown using the model presented below). For each VL, the reserved bandwidth becomes $(l^{\max} + 20) \times 8/\text{BAG}$=3.076Mbps. In this configuration, one source ES can manage at most $\lfloor 100/3.076 \rfloor$=32 VLs aggregating in total 96 Sub-VLs. Therefore, without any additional hardware, the processing capability of the source ES can be improved by around 48%, leading to a better bandwidth utilization.

4.2.2 Computation of the BAG of Aggregated Flows and the Delay due to Sub-VL Aggregation

Consider the aggregation of n Sub-VLs, $1 \leq n \leq 4$, into one VL. Each Sub-VL$_i$ is characterized by its minimum time interval T_i and MFS $l_i^{\max}$. The frame rate of Sub-VL$_i$ is bounded by

$\rho_i = 1/T_i$. Then the maximum arrival frame rate (AFR) of Sub-VLs in a VL is $\rho = \sum_{i=1}^{n} \rho_i$. Let $L_{\max}$ be the MFS of VL:

$$L_{\max} = \max_{1 \le i \le n} \{l_i^{\max}\}. \tag{4.1}$$

Denote by $r = 1/\text{BAG}$ the maximum frame rate in a VL. Obviously, to guarantee that no frame will be blocked due to Sub-VL aggregation, there should be $r \ge \rho$. Moreover, the BAGs must be chosen from the set $\{2^k\}_{k=0}^{7}$ (ms). Therefore, for an appropriate bandwidth allocation, the BAG should be the one with maximum value that meets all the constraints, that is:

$$\text{BAG} = \max_{k=0,\ldots,7} \left\{ 2^k \le \left(\sum_{i=1}^{n} \rho_i \right)^{-1} \right\}, \tag{4.2}$$

which can be expressed equivalently as:

$$\text{BAG} = 2^{\min\left(\left\lfloor \log_2 \left(\sum_{i=1}^{n} \rho_i \right)^{-1} \right\rfloor, 7 \right)}. \tag{4.3}$$

Then the required frame transmission rate (RFTR) for the VL is 1/BAG.

As several Sub-VL queues share the same VL, a frame in a specific Sub-VL$_i$ queue may be delayed due to the RR scheduling. Let D_{SVL_i} be the worst-case queuing delay of Sub-VL$_i$ introduced by Sub-VL aggregation. D_{SVL_i} can be analyzed by using the formulation presented in [59]. Suppose that the Sub-VL is dedicated to one source flow and denoted by |VL| the cardinal number of Sub-VLs belonging to the VL. Let q be the number of packets that are ready for transmission in the Sub-VL$_i$ queue. Sub-VL$_i$ shares VL$_k$ with the other Sub-VLs. Then D_{SVL_i} can be calculated as:

$$D_{SVL_i} = \max_{q=1,2,\ldots} \left[w_i(q) - (q-1)T_i \right], \tag{4.4}$$

where

$$w_i(q) = (q-1)\text{BAG}_k + \sum_{\substack{j \ne i \\ 1 \le j \le |VL_k|}} \left(\left\lfloor \frac{(q-1)T_i}{T_j} \right\rfloor + 1 \right) \text{BAG}_k. \tag{4.5}$$

Obviously, Sub-VL aggregation may introduce extra delay, although it helps improving bandwidth utilization efficiency of VLs. This should be taken into account in network design. The trade-off between traffic load and delay due to Sub-VL aggregation is considered in Section 4.4.

4.2.3 Non-Determinism in VL Transmission

In AFDX networks, the idle state in frame transmission can be introduced by the mismatch between the BAG and the period for periodic source flows or by the arrival uncertainty for sporadic source flows. In either case, frame arrival at destination ESs is undetermined, which might be confused with a fault due to frame loss. To illustrate the issue, in the example shown in Figure 4.2, the frame P3 is assumed to be lost in transmission. The destination ES will not detect the fault until the reception of frame P4, because the destination ES does not know when the next frame will arrive. Under this protocol, the destination ES cannot distinguish between transmission silence and data loss. This is indeed a source of non-determinism.

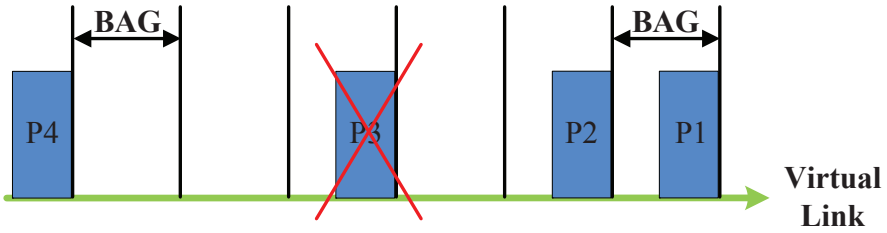


Figure 4.2 Destination End System cannot detect the loss of frame P3 until it receives P4.

Inspired by synchronous transmission schemes, this problem can be solved by inserting frames in source ES to ensure that the VL has one frame and always one frame to transmit in every BAG. With this mechanism, the determinism of frame arrival can be improved. Note that the improvement in terms of determinism is at the expense of a possible average delay increase, as we can observe in synchronous transmission schemes, such as TDMA. Note also that as frame insertion is performed at VL level, there is no impact on the worst-case performance. Nevertheless, this operation will increase the actual network load. Therefore, it is of practical interest to minimize the number of inserted frames via appropriate Sub-VL aggregation schemes.

4.3 Determinism Enhancement with Frame Insertion

In this section, we address the mechanism suggested for frame insertion in VLs to tackle the non-determinism issue related to frame arrival uncertainty. Sub-VL aggregation is then incorporated in this mechanism to mitigate load increase due to frame insertion. We also calculate the required bandwidth after frame insertion and formulate a measure for load increases. These formulations will be used in the resolution of the multi-objective optimization problem studied in Section 4.4.

4.3.1 Frame Insertion in VL

To make transmission deterministic, filler frames are inserted to guarantee that the source ES sends a frame in every BAG. As shown in Figure 4.3, an empty flag and a multiplexer are introduced to implement such a mechanism. If it is the time for transmission and there is no frame in VL FIFO, the empty flag is triggered. Then the multiplexer takes a filler frame from filler frame controller and forwards it into the VL sequence. Otherwise, the multiplexer outputs the data frame from the VL FIFO queue. After every transmission, the multiplexer is halted until the end of the current BAG. Since frame insertion may be needed only if the VL FIFO queue is empty, the VLs are in general not strictly periodic, which is in accordance with the expected behavior of VLs in AFDX networks. Note that, depending on the level of criticality required by specific applications, frame insertion may be performed with a predefined interval bigger than 1 BAG. This would allow reducing the network load while still ensuring determinism. Nevertheless, the formulation presented below can be adapted to this case.

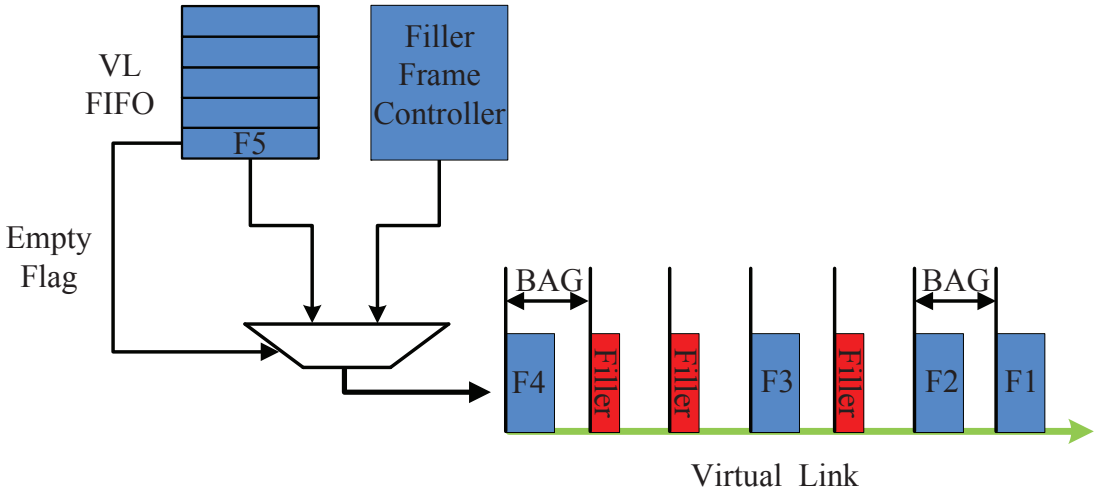


Figure 4.3 Proposed mechanism for enhancing the determinism of AFDX networks.

In this mechanism, the size of the filler frame L_{filler} can be 64bytes, the minimum value specified in the AFDX protocol. This would ensure that the filler frame will have no impact on the MFS of the VL. Furthermore, it is obvious that frame insertion will not change the BAG of the VL. As BAG and MFS are the parameters of a regulated VL utilized in worst-case performance analysis, frame insertion in a VL stream has no impact on the worst-case end-to-end delay of the VL, D_{worst}, measured from the VL regulator to the destination ES. Note that the best case end-to-end delay of the VL, D_{best}, is the sum of technology latencies and transmission time, which is determined by the route and MFS of each VL.

Since the filler frame has no impact on the route and MFS, D_{best} is unchanged with frame insertion. With filler frame insertion in source ES, it is ensured that there is one frame departing from source ES within every BAG of each VL. Therefore, it is guaranteed that when the destination ES finishes one reception, it must receive another frame within the time interval, $[(\text{BAG} - D_{worst} + D_{best})^+, \text{BAG} + D_{worst} - D_{best}]$ as shown in Figure 4.4. By notation, $(x)^+ := \max(x, 0)$.

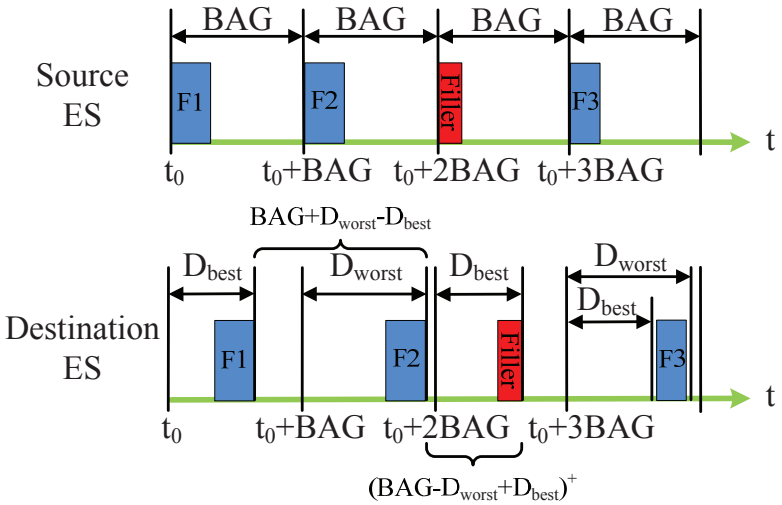


Figure 4.4 Reception time interval in destination ES with frame insertion.

Nevertheless, filler-frame-based determinism enhancement is achieved at the expense of network load increase. In the follows, we try to mitigate this problem by Sub-VL aggregation.

4.3.2 Frame Insertion Based on Sub-VL Aggregation

To optimize network load, it is possible to aggregate several source data flows into a single VL, thus limiting the number of filler frames. As illustrated in Figure 4.5, the Sub-VL FIFO queues are read into VL FIFO with a RR sequence and then a possible frame insertion follows. If a frame either from the Sub-VLs or from the filler frame controller is sent to the VL, the multiplexer is halted until the BAG ends.

Let us recall that a VL is characterized by two main parameters: the MFS and the BAG computed from (4.1) and (4.3), respectively.

4.3.3 Bandwidth Requirement with Frame Insertion

Suppose that Sub-VL$_i$ has the frame rate ρ_i and the MFS $l_i^{\max}$. If a VL is formed by only one Sub-VL, then the frame rate of VL is $r = 1/\text{BAG}$, and with frame insertion the AFR in

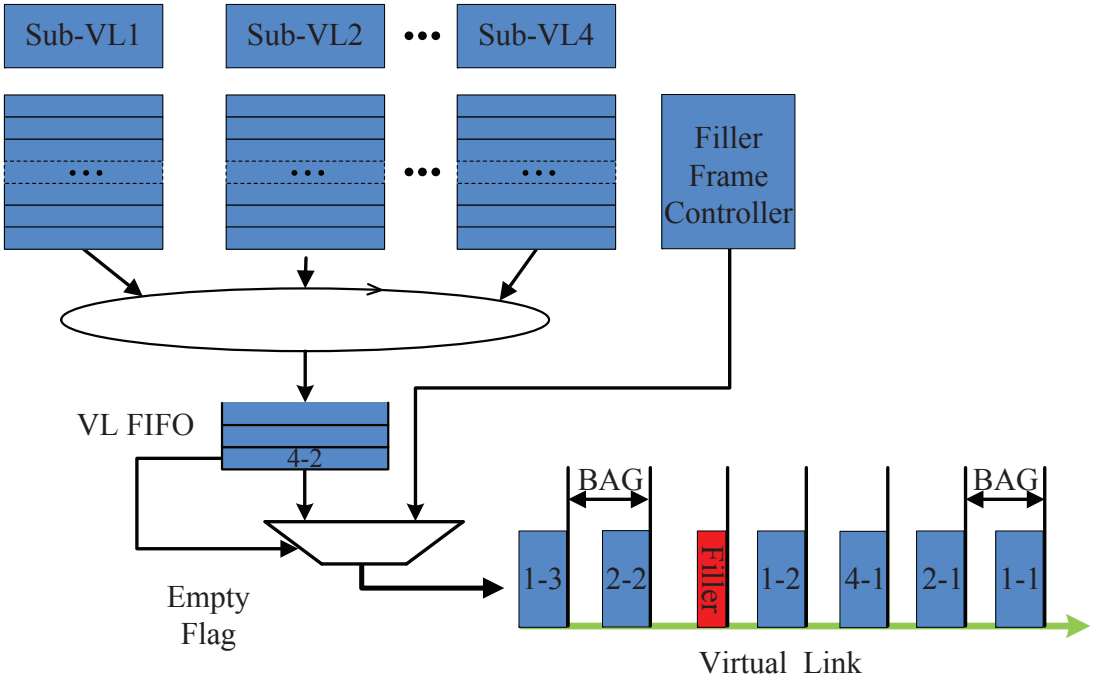


Figure 4.5 Frame insertion based on Sub-VL aggregation.

total is $\rho = \rho_i$. Therefore, the rate difference, $r - \rho$, denotes the number of inserted frames per time unit. Then the required bandwidth is given by:

$$BW = l_i^{\max} \times \rho_i + L_{filler} \left(\mathrm{BAG}^{-1} - \rho_i \right), \tag{4.6}$$

and the reserved bandwidth is $l_i^{\max}/BAG$.

If a VL aggregates n Sub-VLs, the AFR in total is $\rho = \sum_{i=1}^{n} \rho_i$. Therefore the required bandwidth is:

$$BW = \sum_{i=1}^{n} (l_i^{\max} \times \rho_i) + L_{filler} \left(\mathrm{BAG}^{-1} - \sum_{i=1}^{n} \rho_i \right), \tag{4.7}$$

and the reserved bandwidth is $L_{\max}/BAG$.

The second part on the right-hand side of (4.6) and (4.7) is the bandwidth increase after frame insertion, which is related to the rate difference, $r-\rho$. Hence, one objective for optimal Sub-VL aggregation is to minimize the bandwidth increase.

4.4 Optimal Sub-VL Aggregation

In order to prevent buffer overflows in the ESs, the RFTR of a VL has to be higher or equal to the AFR of the Sub-VLs it carries. In the case where the arrival period of the source falls into $\{2^k\}_{0\leq k\leq 7}$, the AFR and the RFTR for VL will be identical. Otherwise, frame insertion is required. Either way, there is no idle BAG for VL and the rate difference is equivalent to the load increase in unit time. Hence, an appropriate scheme for Sub-VL aggregation is the one that allows minimizing the sum of rate difference for all VLs, so that the overhead due to frame insertion is the minimum. Meanwhile the delay introduced by Sub-VL aggregation should also be optimized. Therefore, the Sub-VL aggregation should be modeled as a multi-objective optimization problem.

Essentially, there are three main constraints in Sub-VL aggregation:

— Sub-VLs for aggregation should have the same source and destination ESs.

— A VL contains at most 4 Sub-VLs.

— The sum of AFR after Sub-VL aggregation cannot exceed $r_{\max}$ (1K frame/s).

4.4.1 Formulation of Optimal Sub-VL Aggregation

Let $S = \{\text{Sub-VL}_1, \text{Sub-VL}_2, ..., \text{Sub-VL}_N\}$ be a set of N Sub-VLs. Denote by ρ_i the frame rate of Sub-VL_i. For all Sub-VLs, AFR in total is $\rho_0 = \sum_{i=1}^{N} \rho_i$.

Let $\text{VL}_j = \{\text{Sub-VL}_{j_1}, \ldots, \text{Sub-VL}_{j_n}\}$, $\forall \text{Sub-VL}_{j_k} \in S$, $k = 1, \ldots, n$. We can then formulate the Sub-VL aggregation problem as the partitioning of S into m non-empty VLs with $[\lceil N/4 \rceil \leq m \leq N]$.

Let $\mathcal{P} = \bigcup_{i\in\mathcal{I}} \mathcal{P}_i$ be the set of all admissible partitions, where $\mathcal{I}$ is the index set. In combinatorics, the number of possible partitions in which a set of N elements can be split is bounded by the so-called Bell number [3], denoted B_N. However, the partitioning of Sub-VLs has other constraints such as subset size. Hence, the cardinal of $\mathcal{P}$ may be significantly smaller than B_N. Then the ith partition can be denoted by

$$\mathcal{P}_i := \left\{\text{VL}_1^i, \text{VL}_2^i, \ldots, \text{VL}_j^i, \ldots, \text{VL}_m^i\right\} \vdash S, \tag{4.8}$$

where VL_j^i is an aggregation of Sub-VLs and the symbol " $\vdash$ " stands for "is a partition of." Denote by BAG_j^i the BAG of VL_j^i. For VL_j^i, the RFTR is $r_j^i = 1/\text{BAG}_j^i$. According to (4.3),

r_j^i is given by:

$$r_j^i = \frac{1}{\mathrm{BAG}_j^i} = 2^{-\min\left(\left\lfloor \log_2\left(\sum_{1\le k\le \left|\mathrm{VL}_j^i\right|} \rho_k\right)^{-1} \right\rfloor, 7\right)}, \tag{4.9}$$

where ρ_k is the frame rate of Sub-VL$_k$ belonging to VL_j^i. The RFTR R_i for the partition $\mathcal{P}_i$ is

$$\begin{aligned} R_i &= \sum_{\mathrm{VL}_j^i \in \mathcal{P}_i} r_j^i \\ &= \sum_{\mathrm{VL}_j^i \in \mathcal{P}_i} 2^{-\min\left(\left\lfloor \log_2\left(\sum_{1\le k\le \left|\mathrm{VL}_j^i\right|} \rho_k\right)^{-1} \right\rfloor, 7\right)}. \end{aligned} \tag{4.10}$$

The rate difference, y_i, representing the wasted bandwidth, can be expressed as:

$$\begin{aligned} y_i &= R_i - \rho_0 \\ &= \sum_{\mathrm{VL}_j^i \in \mathcal{P}_i} 2^{-\min\left(\left\lfloor \log_2\left(\sum_{1\le k\le \left|\mathrm{VL}_j^i\right|} \rho_k\right)^{-1} \right\rfloor, 7\right)} - \rho_0. \end{aligned} \tag{4.11}$$

Obviously, R_i varies according to the partition, and so does y_i. To reduce the load increase, we need to minimize the rate difference y_i in (4.11). Note that for a given set of Sub-VLs, ρ_0 is a constant. Hence, minimizing the rate difference is equivalent to obtain the minimum R_i.

The delays introduced by Sub-VL aggregation can be measured in different ways that will influence the optimization procedure. For example, if the maximum worst-case delay among all the VLs is chosen as the cost function, then the result will tend to be the lowest allowed delay for all the Sub-VLs. A less pessimist choice is the use of the average worst-case delay of all the Sub-VLs:

$$D_{\mathcal{P}_i} = \frac{1}{N} \sum_{\mathrm{VL}_j^i \in \mathcal{P}_i} \sum_{1\le k\le \left|\mathrm{VL}_j^i\right|} D_{SVL_k}, \tag{4.12}$$

which is the second cost function considered in the resolution of optimal Sub-VL aggregation problems in the present work.

Let $G_1(\mathcal{P}_i) = R_i$ and $G_2(\mathcal{P}_i) = D_{\mathcal{P}_i}$. Optimal Sub-VL aggregation amounts then to solving

the following multi-objective optimization problem:

$$\min_{\mathcal{P}_i \in \mathcal{P}} G(\mathcal{P}_i) = [\, G_1(\mathcal{P}_i), G_2(\mathcal{P}_i) \,]^T ; \tag{4.13}$$

$$\text{s.t. : } 1 \leq i \leq B_N; \tag{4.14}$$

$$\lceil N/4 \rceil \leq j \leq N; \tag{4.15}$$

$$1 \leq k \leq 4; \tag{4.16}$$

$$\sum_{1 \leq k \leq \left|\mathrm{VL}_j^i\right|} \rho_k \leq r_{\max}; \tag{4.17}$$

where i is the partition index, N is the total number of Sub-VLs, and k represents the index of a Sub-VL in a VL. In AFDX networks, the sum of AFRs of the aggregated Sub-VLs cannot exceed $r_{\max}$ (1K frame/s), the maximum rate of a single VL.

Since Sub-VL aggregation can introduce extra delay, an optimal solution to this problem can only be achieved in the sense of Pareto by considering the possible trade-off between these two objectives (see, e.g., [91]).

4.4.2 Lexicographic Method for Optimal Sub-VL Aggregation

In order to find a Pareto optimal solutions, system designers should impose design preferences [91]. In the considered problem, the primary objective is to reduce the load increase based on which we will try to minimize the delay introduced by Sub-VL aggregation. The lexicographical optimization method is suitable for this setup. More precisely, minimizing the rate difference is solved first. Then, a δ-constraint is imposed to control the trade-off between load increase and the delay due to Sub-VL aggregation. The corresponding lexicographic formulation can be given as follows:

$$\min_{\mathcal{P}_i \in \mathcal{P}} \; G_l(\mathcal{P}_i); \tag{4.18}$$

$$\text{s.t. : } (4.14) - (4.17);$$

$$G_1(\mathcal{P}_i) \leq (1+\delta)\, G_1(\mathcal{P}_1^*), \text{ for } l = 2; \tag{4.19}$$

$$l = 1, 2;$$

where $\mathcal{P}_1^*$ represents the optimum of the first objective function. δ is a nonnegative value that can be varied to tighten the constraint. Note the multi-objective functions are solved in sequence to find the Pareto optimal points [91].

To illustrate the main property of the above multi-objective optimization problem, we con-

sider an example of 8 Sub-VLs having different periods as shown in Table 4.1. By using exhaustive enumeration, all possible solutions regarding the first objective can be obtained as shown in Figure 4.6. Meanwhile, we can construct the so-called Pareto front for this multi-objective optimization problem (see Figure 4.7). Note that in Figure 4.7, many points are overlapped in R_i-$D_{\mathcal{P}_i}$ plane.

Table 4.1 Parameters of Sub-VLs

Sub-VL	1	2	3	4	5	6	7	8
Period (ms)	10	25	30	40	60	80	100	125

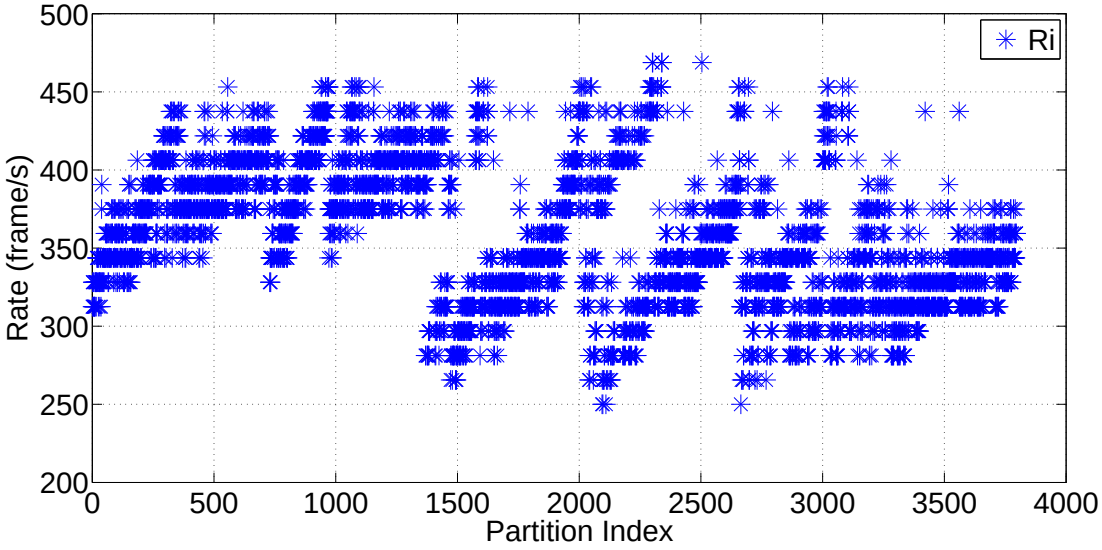


Figure 4.6 RFTR with parameters in Table 4.1.

For $\delta = 0$, the point with an RFTR of 250 frame/s and an average worst-case delay of 22ms will be the Pareto optimal solution, which has the minimum RFTR among all possible solutions, and the delay introduced by Sub-VL aggregation for the partition is the smallest under the δ-constraint.

If we relax the δ-constraint, for example let $\delta = 20\%$, then more partitions with RFTR$\leq$ 300 frame/s can be included in the set of candidate solutions. In this case, the solution point with (296.9, 6) is found to be the Pareto optimal solution. In this case, the introduced delay is reduced at the expense of network load increase. More details about this example are discussed later in Section 4.5.2.

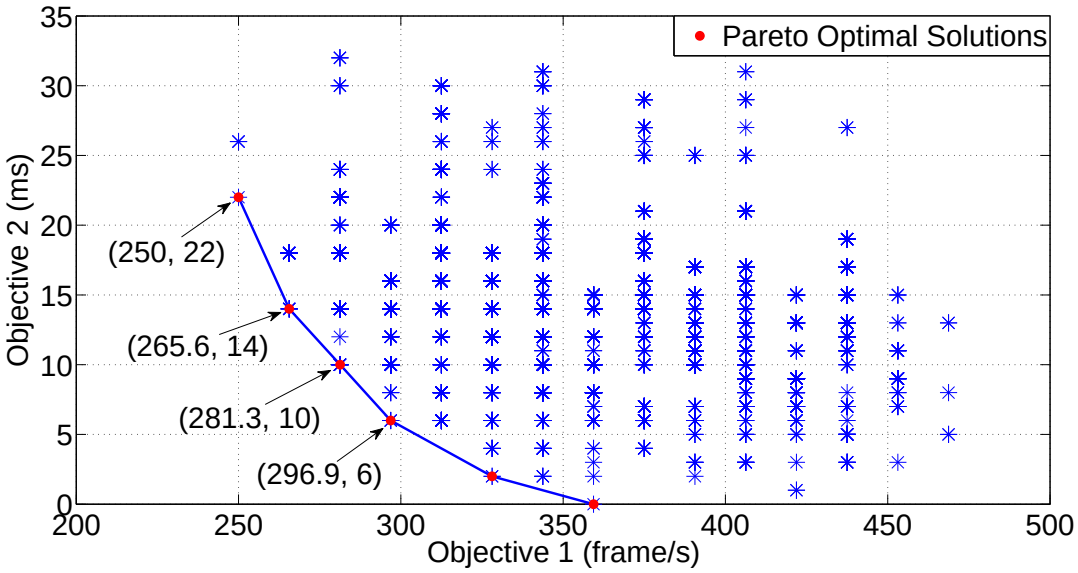


Figure 4.7 All possible solutions and Pareto front.

4.4.3 Algorithms for Sub-VL Aggregation

The multi-objective optimization problem can be solved in two iterations. The first iteration is to obtain $\mathcal{P}_1^*$ with the minimum RFTR $\mathcal{R}^*$. The second iteration is to find the partition with minimum delay introduced by Sub-VL aggregation under (4.14)-(4.17) and (4.19). According to (4.10), R_i is a discrete and nonlinear function with respect to the partition. Therefore, the objective function $G_1(\mathcal{P}_i)$ is not convex and might not admit a unique global minimum, as shown in Figure 4.6. Similarly, $G_2(\mathcal{P}_i)$ is also nonlinear and non-convex. Therefore, the solution for the multi-objective optimization problem might not be unique. Indeed the considered problem is a special multiple knapsack problem, which is NP-Complete. Furthermore, it is very different from standard multiple knapsack problems when considering the trade-off between load increase and the delay introduced by Sub-VL aggregation. In this paper, three algorithms are applied to solve the multi-objective optimization problem.

Brute Force Algorithm

One possible and accurate method to obtain the optimal solution is the brute force algorithm. The optimal solution can be found by an exhaustive enumeration of all solutions. In the first iteration, an optimal solution $\mathcal{P}_1^*$ that leads to the minimum rate difference is obtained. Then by adding the δ-constraint in the second iteration, we can get a global optimal result for the multi-objective optimization problem. However the computational complexity grows exponentially with the number of Sub-VLs. Note that more efficient algorithms, such as

branch-and-bound algorithm [97] and the greedy algorithm [42] [121] [122], can be explored to address the optimization problem with large size, when applying the proposed formulation to real-life applications. In present work, we consider to use the greedy algorithm, which is well known and computationally efficient.

Greedy Algorithm

The strategy behind the greedy algorithm is to make local optimal choices at every step of each iteration with the hope of finding a globally optimal result. The heuristics used in the developed algorithm is that for each step, the optimization strategy is to select one aggregation of Sub-VLs, through which R_i or $D_{\mathcal{P}_i}$ can be reduced the most. Let $\mathcal{V}$ be the candidate set of VLs composed of 2 to 4 Sub-VLs. Suppose that $\forall v \in \mathcal{V}$, the corresponding RFTR does not exceed the rate limit $r_{\max}$. Since the Sub-VLs cannot be reused, all VLs, containing one or more selected Sub-VLs, are removed from the candidate set at the end of each step. The greedy algorithm stops and gives the local optimal solution when the candidate set is empty.

As an example, we consider a set of 3 Sub-VLs characterized by their periods: {6ms, 20ms, 40ms}. The candidate set $\mathcal{V}$ is {{6ms, 20ms}, {6ms, 40ms}, {20ms, 40ms}, {6ms, 20ms, 40ms}}. For the subset {6ms, 20ms}, the RFTRs before and after aggregation are 312.5 frame/s and 250 frame/s, respectively. Therefore, the reduction after aggregation is 62.5 frame/s. Furthermore, let us denote by $D_v = \sum_{1 \leq i \leq |v|}(D_{SVL_i})$ the total introduced worst-case delay of the Sub-VLs in one VL. In this case, D_v is 8ms. Accordingly, the rate difference and the introduced delay for the other aggregations are calculated and listed in Table 4.2.

Table 4.2 Sub-VL Aggregation Candidates

Subsets in $\mathcal{V}$ (ms)	RFTR without Aggregation(frame/s)	RFTR (frame/s)	Δ (frame/s)	D_v (ms)
{6, 20}	312.5	250	62.5	8
{6, 40}	281.25	250	31.25	8
{20, 40}	93.75	125	-31.25	16
{6, 20, 40}	343.75	250	93.75	24

In the first iteration, the aggregation of {6ms, 20ms, 40ms} is selected in the first step and all the VLs comprising these Sub-VLs are removed from the candidate set. Then the greedy algorithm stops as the candidate set becomes empty. In the second iteration, the candidate set is sorted first in ascending order for the delay introduced by Sub-VL aggregation, and then by descending order for the reduction. Suppose that all subsets in $\mathcal{V}$ are allowed with

the δ-constraint. Therefore, {6ms, 20ms} with a total delay of 8ms and a reduction of 62.5 frame/s is selected. Consequently, all the candidates are removed from the candidate set because they share one or more selected Sub-VLs. Then the greedy algorithm stops as the candidate set becomes empty. In this case, the local optimal solution is {{6ms, 20ms}, {40ms}}. The rigorous formulation of the developed greedy algorithm is described below.

Algorithm 1 Greedy Algorithm

Input: S, δ;

Output: $(R_{\min}, D_{\min})$ and corresponding partition;

Initial: $F_1 = \varnothing,\ S_1' = S,\ F_2 = \varnothing,\ S_2' = S$;

Construct $\mathcal{V}$, the set of all possible VLs;
For each $v \in \mathcal{V}$, compute the gain $\Delta(v)$ and the extra delay D_v;
Discard the VLs with the negative or null gain;
Sort the VLs by decreasing $\Delta(v)$ and insert them in a list L_1;
Sort the VLs first by ascending D_v and then by decreasing $\Delta(v)$ and insert them in a list L_2;
while $L_1 \neq \varnothing$ **do**
    Add $L_1[1]$ (the first VL with biggest gain in current L_1) to the solution F_1, $S_1' = S_1' \cap L_1[1]$;
    Remove all VLs from L_1 which contain any Sub-VL of $L_1[1]$, including $L_1[1]$ itself;
end while
Obtain $\mathcal{P}_1^* = F_1 \cup S_1'$ and then calculate $\mathcal{R}^*$;
while $L_2 \neq \varnothing$ **do**
    if $\frac{r_a(L_2[1])}{\sum_{1 \leq i \leq |L_2[1]|} \rho_i} \leq (1+\delta)\frac{\mathcal{R}^*}{\sum_{1 \leq i \leq |S|} \rho_i}$ **then**
        Add $L_2[1]$ to the solution F_2, $S_2' = S_2' \cap L_2[1]$;
        Remove all VLs from L_2 which contain any Sub-VL of $L_2[1]$, including $L_2[1]$ itself;
    else
        Remove $L_2[1]$ from L_2;
    end if
end while
Obtain $\mathcal{P}_{\min} = F_2 \cup S_2'$ and then calculate $(R_{\min}, D_{\min})$;
Output $(R_{\min}, D_{\min})$ and $\mathcal{P}_{\min}$.

We define a function $\Delta : \mathcal{V} \to \mathbb{R}$ which gives, for each VL, the gain obtained with the aggregation of its Sub-VLs. Denoting by $r_b(v)$ the sum of the frame rate required by the Sub-VLs within a VL before aggregation and by $r_a(v)$ the RFTR after aggregation, we have

for all $v \in \mathcal{V}$:

$$\begin{aligned}\Delta(v) &= r_b(v) - r_a(v) \\ &= \sum_{1 \le i \le |v|} 2^{-\min\left(\left\lfloor \log_2 \rho_i^{-1} \right\rfloor, 7\right)} \\ &\quad - 2^{-\min\left(\left\lfloor \log_2 \left(\sum_{1 \le i \le |v|} \rho_i\right)^{-1} \right\rfloor, 7\right)}.\end{aligned} \tag{4.20}$$

Note that the gain $\Delta(v)$ can be either positive, negative or null depending on the Sub-VLs in v. We define a subset of $\mathcal{V}$, $F \subset \mathcal{V}$, such that $\forall v_1, v_2 \in F$, $v_1 \cap v_2 = \varnothing$. Then the local minimum of the first objective function, $\mathcal{P}_1^*$, can be achieved by maximizing the gain, $\sum_{v \in F} \Delta(v)$. Meanwhile, we can compute the total worst-case delay of the Sub-VLs, D_v.

Then the local optimal solution for the multi-objective optimization problem can be obtained by minimizing the delay introduced under the added δ-constraint in the second iteration. The developed greedy algorithm is summarized in Algorithm 1.

Although the greedy algorithm cannot guarantee to find the global optimum, it is much less time consuming compared to the brute force algorithm. An experiment implemented with Matlab® for a set of 100 Sub-VLs with randomly generated period for all the Sub-VLs shows that the execution can be terminated within minutes. More detailed results of experiments will be presented in Section 4.5.

Greedy Algorithm with Pre-processing

The complexity of the greedy algorithm is related to the search space size. It happens that when grouping some Sub-VLs together, the equivalent period is a power of 2. In this case, the search space can be reduced if we perform these aggregations first and remove them from set of Sub-VLs.

For example, the periods of Sub-VL_1, Sub-VL_2 and Sub-VL_3 are, respectively, 5ms, 5ms and 10ms. To aggregate these three Sub-VLs into one VL, the equivalent period is 2ms. According to (4.3), the BAG for this VL is 2ms. In this case, no filler frame insertion is needed.

Based on the above analysis, a greedy algorithm with pre-processing is developed. This algorithm has two steps. The first step is to find special cases mentioned above that perfectly fill VLs among the Sub-VL set. The second step is the use of the greedy algorithm in Section 4.4.3 to find the local optimal result with the reduced Sub-VL set.

In summary, the brute force algorithm can reach the global optimal solution, but it is suitable

only for small number of considered Sub-VLs. The greedy algorithm and its variation with pre-processing are suitable for large size problems, although they may lead to local optimums. The performance of these algorithms are illustrated in the next section.

4.5 Performance Evaluation

In this section, we perform numerical simulations of the proposed mechanism to verify its feasibility. Then the performance of different optimization algorithms for Sub-VL aggregation is evaluated using different configurations.

4.5.1 Validation of Frame Insertion Mechanism

In Section 4.3.2, a mechanism with frame insertion based on Sub-VL aggregation is put forward. The feasibility of such a mechanism is verified in this section by numerical simulations using Matlab® and TrueTime [30]. In the considered example, there are three Sub-VLs whose parameters are listed in Table 4.3. Based on (4.3), the BAG for the aggregated VL is 4ms.

Table 4.3 Parameters of Sub-VLs

Sub-VL	Period (ms)	Max Jitter (ms)	Frame Size (byte)
Sub-VL1	10	3	80
Sub-VL2	60	18	180
Sub-VL3	25	7.5	130

First, Matlab® is used to simulate the aggregation and regulation in source ES. In this simulation, we use unit frame size. The simulation results are shown in Figure 4.8.

The jitter of each Sub-VL is considered in the simulation. Due to the jitter, the accurate positions of filler frames cannot be determined in advance. As shown in Figure 4.5, the empty flag acts as a trigger. When there is nothing from Sub-VLs to transmit during the period of 1 BAG, the empty flag is triggered. Then the filler frame is forwarded into VL. The simulation result is shown in Figure 4.8. The red line with "$*$"-mark indicates the inserted frames. The other three different colors and shapes represent the Sub-VLs. The frames of Sub-VLs and inserted frames are regulated according to the BAG. They are transmitted every 4ms. According to the simulation results, the proposed mechanism is feasible.

Furthermore, a TrueTime simulation is set up to implement the proposed mechanism. TrueTime is a more accurate timing simulator based on Matlab®/Simulink, which can model data transmission using different network protocols and task execution in real-time kernels [30]. A simple AFDX network shown in Figure 4.9 is set up. Sub-VL aggregation and frame

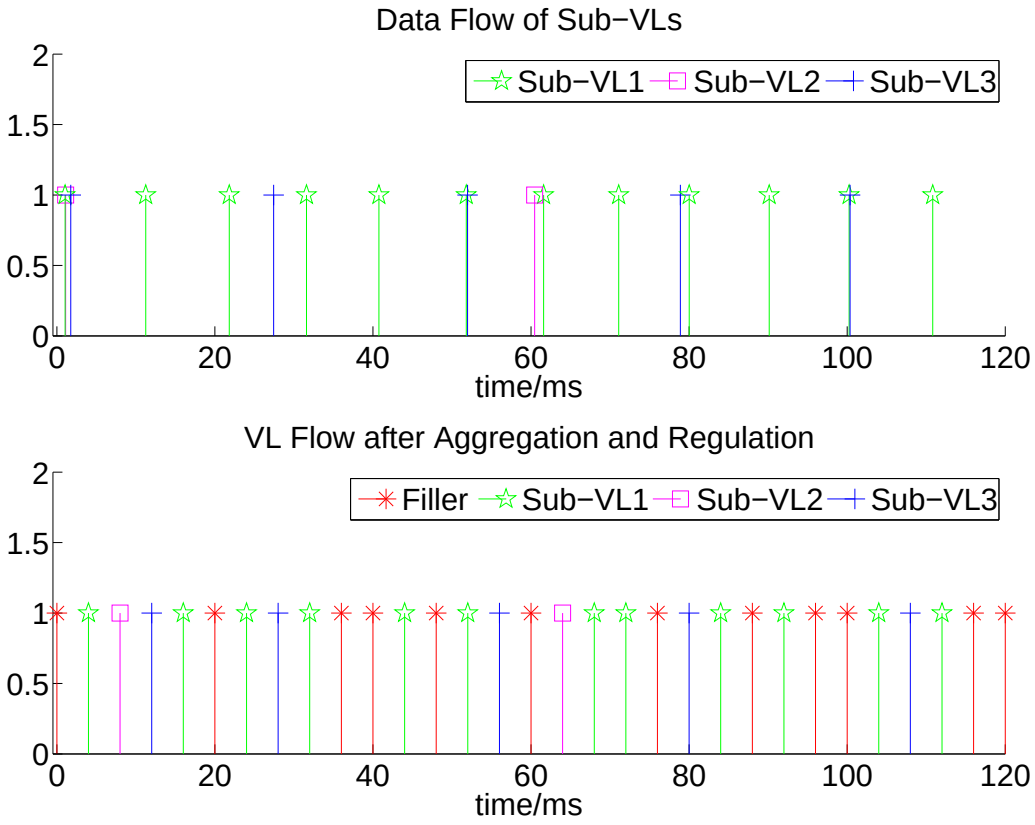
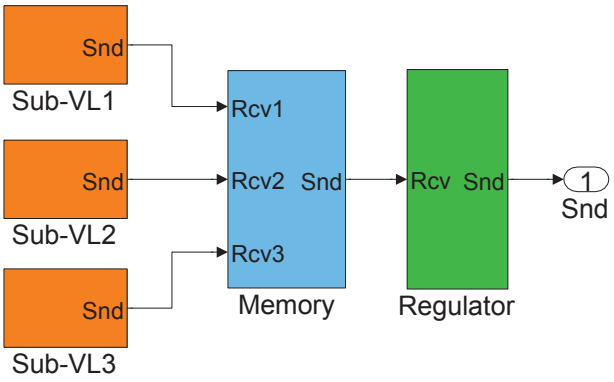


Figure 4.8 Matlab® simulation result of frame insertion based on Sub-VL aggregation.

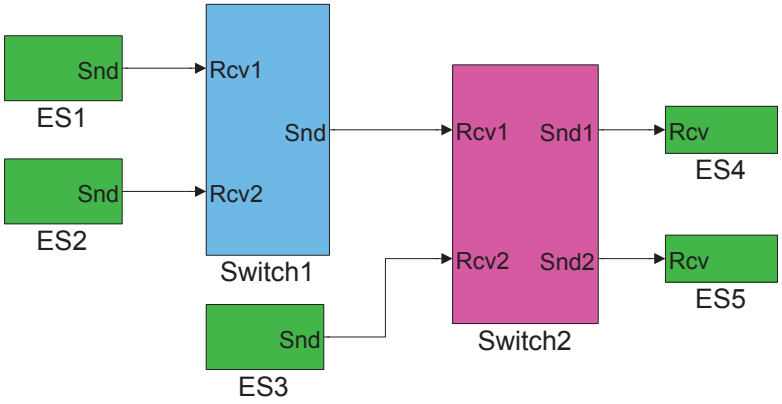
insertion are added into the model of ES1. The simulated system can perform a real-time data transmission.

After Sub-VL aggregation, frame insertion, and VL regulation, all frames are forwarded into a VL sequence. Frame insertion function is implemented in the VL regulator model. The simulation results of Sub-VL aggregation are shown in Figure 4.10. The data flow without insertion is also presented for comparison. As stated in Section 4.3.1, we can guarantee the determinism of frame arrival with frame insertion. The real-time simulation results confirms the feasibility of such a mechanism.

Using this structure, the end-to-end delay analysis can be performed. In addition, real-time fault detection can also be executed. We can set a probability of data loss in switches. With the expected deterministic reception on destination ESs, it is easy to detect some classes of faults such as lost frames in real-time. Furthermore, this AFDX network simulation system is extensible to more complex network configurations, which allows carrying out additional fault injection and fault analysis.



(a) Sub-VL aggregation and VL regulation model



(b) AFDX simulation system configuration

Figure 4.9 AFDX simulation system based on TrueTime.

4.5.2 Evaluation of Sub-VL Aggregation Strategies

In the follows, comparison of different aggregation schemes is performed to evaluate the developed optimal Sub-VL aggregation strategies.

The considered system is a network with 8 Sub-VLs studied in Section IV.B whose parameters are given in Table 4.1. According to (4.7), L_{filler} is smaller or equal to $l_i^{\max}$. Hence, the load increase percentage measured in frame is equal or greater than the load increase measured in bit. In this study, the traffic increase is measured as an increased frame rate. In this example, the AFR is a total ρ_0 of 245.5 frame/s. If every Sub-VL is transmitted by one VL, the RFTR is 359.4 frame/s. As there is no aggregation, the introduced delay is zero. Whereas, the load increase is about 46.4% compared to the arrival frames after frame insertion.

Note that some Sub-VL aggregation schemes may lead to poor performance. For example, in the above considered problem, the partition, VL_1={Sub-VL1, Sub-VL2, Sub-VL6, Sub-VL8}

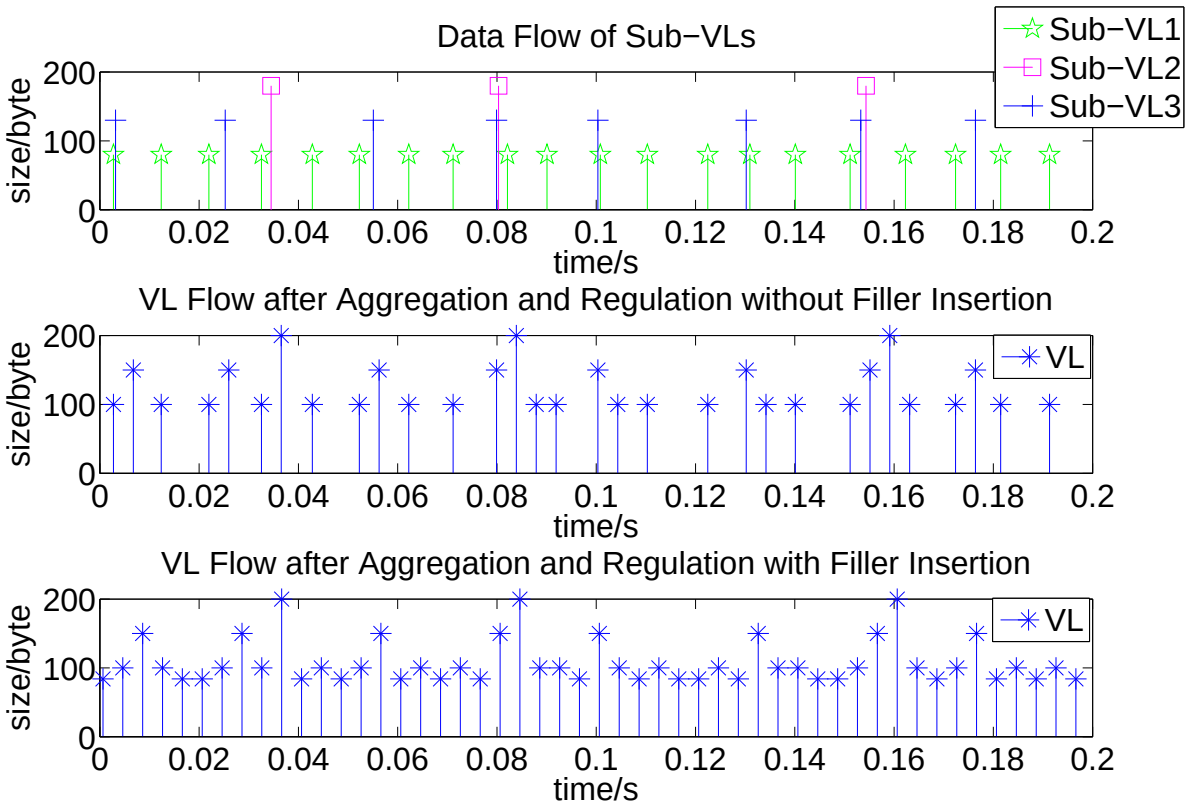


Figure 4.10 Simulation results produced with TrueTime.

and VL_2={Sub-VL3, Sub-VL4, Sub-VL5, Sub-VL7}, will result in as much as 52.75% frame rate increase and 18ms introduced average delay.

To solve the multi-objective optimization problem, the brute force Sub-VL aggregation strategy is applied, in which all admissible partitions are traversed to obtain the global optimal solution under the δ-constraint. In the considered example, the Pareto optimal solution with the constraint of δ=20% is given in Table 4.4. The overall load increase is about 20.9%, which is better than a 46.4% load increase observed when no aggregation is performed and filler packets are inserted. As presented in Section 4.4.2, the introduced average delay, 6ms, is minimal for the partitions under the δ-constraint.

When the greedy algorithm is used in this example, the search is terminated by a local optimal solution. When δ=20%, the local optimal partition is {{Sub-VL1, Sub-VL4}, {Sub-VL2, Sub-VL5}, {Sub-VL3, Sub-VL6}, {Sub-VL7}, {Sub-VL8}}. The corresponding performance is given in Table 4.5. In this case, the load increase is only 14.56%, which is much better than the scheme without Sub-VL aggregation. Furthermore, the introduced average delay is 10ms, It can be observed from Figure 4.7 that this solution is Pareto optimal. However, the greedy

Table 4.4 Performance obtained with the brute force algorithm

Sub-VL Aggregations	AFR (frame/s)	RFTR (frame/s)	Excess Frame Rate in Percent	D_v (ms)
{1, 4}	125	125	0	16
{2, 5}	56.7	62.5	10.29%	32
{3}	33.3	62.5	87.5%	0
{6}	12.5	15.6	25%	0
{7}	10	15.6	56.25%	0
{8}	8	15.6	95%	0
Total	245.5	296.9	20.94%	48

algorithm cannot guarantee to find the Pareto optimal solution. It obtains a local optimal solution for the optimization problem considering the trade-off between the two objectives.

Table 4.5 Performance of the greedy algorithm

Sub-VL Aggregations	AFR (frame/s)	RFTR (frame/s)	Excess Frame Rate in Percent	D_v (ms)
{1, 4}	125	125	0	16
{2, 5}	56.7	62.5	10.29%	32
{3, 6}	45.8	62.5	36.46%	32
{7}	10	15.6	56.25%	0
{8}	8	15.6	95%	0
Total	245.5	281.3	14.56%	80

It can be observed that in this example, the period of a VL aggregating Sub-VL1 and Sub-VL4 is the power of 2. The situation is the same when we aggregate Sub-VL2, Sub-VL6 and Sub-VL7. We can then apply the greedy algorithm with pre-processing. In the first step, we get a reduced set listed in Table 4.6. In the second step, the greedy algorithm is executed over the reduced set. When δ=20%, a local optimal solution of {{Sub-VL1, Sub-VL4}, {Sub-VL2, Sub-VL6, Sub-VL7}, {Sub-VL3, Sub-VL5}, {Sub-VL8}} is obtained. The load increase and the introduced average delay for this partition are 265.6 frame/s and 18ms, respectively. The greedy algorithm with pre-processing provides system designers with an additional option in network tuning.

Table 4.6 Set obtained by first step of pre-processing greedy algorithm

Sub-VL	3	5	8
Period(ms)	30	60	125
Max Jitter(ms)	9	18	37.5

In order to validate the performance with different configurations, many instances with randomly generated periods in the [1ms, 200ms] interval were analyzed. The δ-constraint is set to 0 and 10%, respectively. The results are shown in Figure 4.11, Figure 4.12 and Figure 4.13. For each parameter set considered, three algorithms are applied to obtain global/local optimal solutions for 1000 instances. The avg./worst/best performances for different algorithms are obtained. Compared with the results without aggregation, the solutions using aggregation strategies are much better with respect to load mitigation, even in the worst case. The overhead load in the network due to frame insertion is reduced. It is worth noting that the brute force algorithm cannot finish in a reasonable time with 50 Sub-VLs. For this case, only the greedy algorithm and the greedy algorithm with pre-processing are applied to find the local optimal solutions. Although the solutions may not be Pareto optimal in a strict sense, they are much better than the scheme without Sub-VL aggregation.

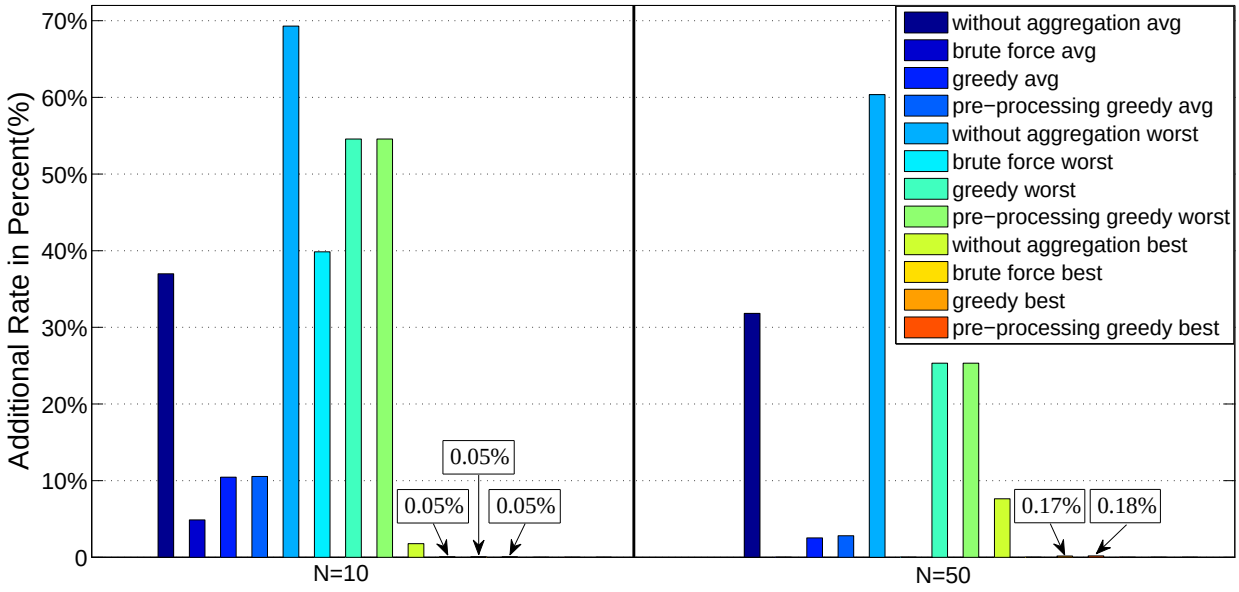


Figure 4.11 Evaluation of the load increase for 10 and 50 Sub-VLs (N=10 and N=50), $\delta = 0$.

4.6 Concluding Remarks

In this paper, a mechanism for frame insertion is proposed to enhance the determinism of AFDX networks with respect to frame arrival uncertainty. In order to reduce the load increase due to frame insertion, a strategy for Sub-VL aggregation is developed, which is formulated as a multi-objective optimization problem considering the trade-off between load increase and the delay introduced by Sub-VL aggregation. Three algorithms are proposed

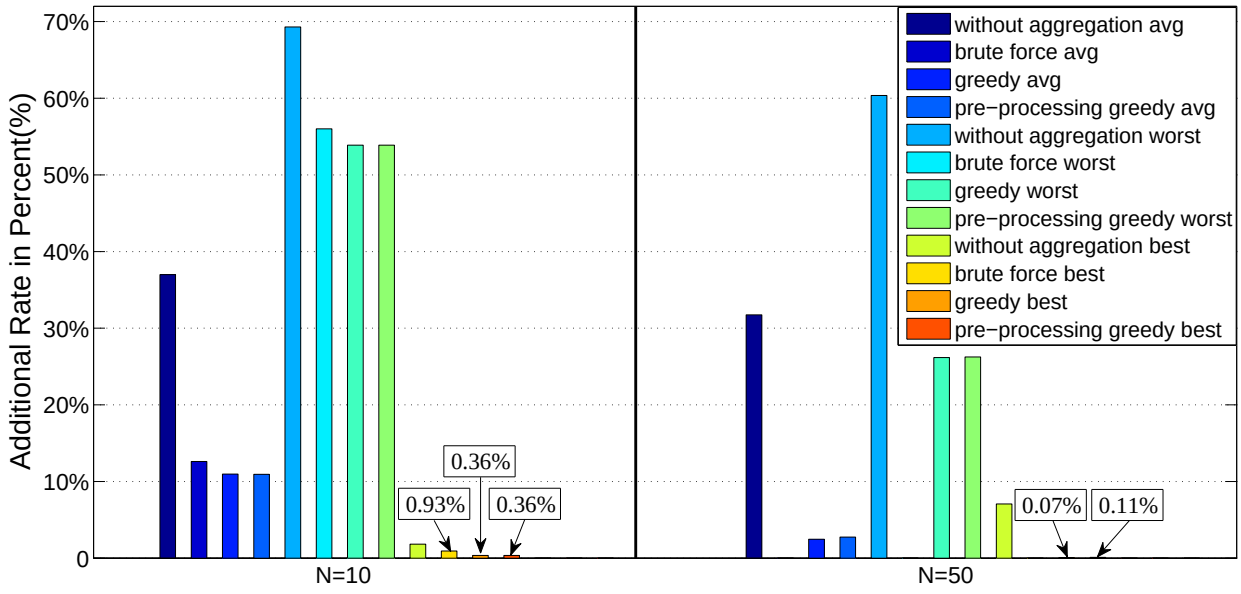


Figure 4.12 Evaluation of the load increase for 10 and 50 Sub-VLs (N=10 and N=50), δ =10%.

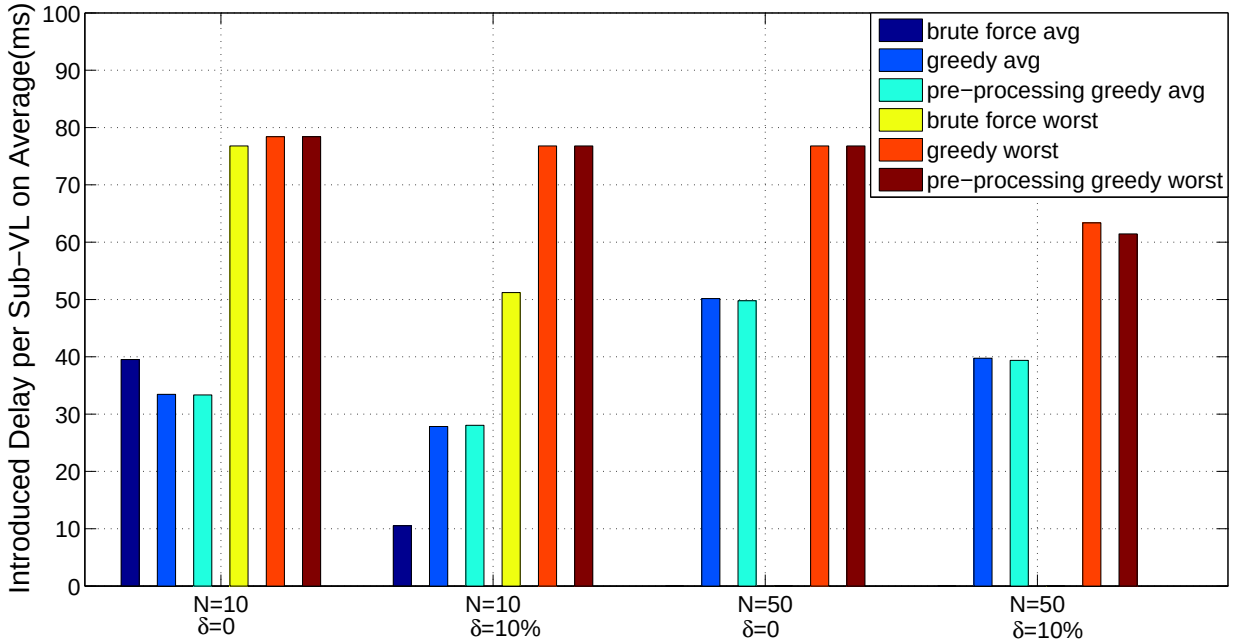


Figure 4.13 Average or worst value of the average delay introduced by Sub-VL aggregation for the specified parameter.

and investigated to solve the Sub-VL aggregation optimization problem. Simulations are carried out to illustrate the feasibility of the proposed filler packet insertion method and to validate the performance of developed algorithms. The results show that the load increase can

be dramatically reduced and the delay introduced by Sub-VL aggregation can be mitigated with a relaxed δ-constraint. Finally, the framework of multi-objective optimization can be extended to incorporate more design considerations.

It is worth noting that the focus of the present work is put on the configuration in source ESs. However the Sub-VL aggregation with frame insertion may have an impact on the entire network. This may raise challenges regarding the practical application of the proposed mechanism. However, it is interesting to note that the work presented in [76] shows that for a case-study composed of a flight management system, the temporal behavior of avionics functions is not significantly affected even when the worst-case network delay has been increased by 400%. Nevertheless, the impact of Sub-VL aggregation with frame insertion has to be carefully evaluated against the overall performance requirements for specific applications in the design of AFDX networks.

4.7 Acknowledgment

The first author holds a scholarship from the China Scholarship Council. This work is sponsored in part by NSERC-CRIAQ CRD project AVIO402, MITACS Acceleration Quebec program, MDEIE-CRIAQ Quebec-China project, and the industrial partners Thales Canada Inc. and Bombardier Aerospace.

CHAPTER 5 ARTICLE 2: INCORPORATING PERFORMANCE ANALYSIS INTO RELIABILITY ASSESSMENT FOR AVIONICS FULL-DUPLEX SWITCHED ETHERNET NETWORKS

In certification of AFDX networks, performance analysis and reliability assessment are two main concerns. In this chapter, an approach is introduced to incorporate performance analysis into reliability assessment. As a result, probabilistic upper bounds, which may be tighter than the deterministic ones, can be applied in AFDX network certification. This can facilitate the performance certification by offering a larger margin regarding delay requirements for delay-sensitive applications. The following sections are based on [84], which has been submitted to Reliability Engineering & System Safety.

Authors—Meng Li, Guchuan Zhu, Michaël Lauer, Yvon Savaria, and Jian Li.

Abstract—AFDX has been developed to meet the challenges due to the growing number of data-intensive applications in modern avionic systems. Although such a network can support high speed data transmission, the jitter due to the inherent asynchronous nature of this protocol is a serious concern affecting its determinism level. In order to certify the timing performance of AFDX networks, the upper bounds with no delay violation are usually computed based on worst-case analysis. However, these upper bounds are in general too pessimistic and the worst-case analysis does not consider the capability of redundant data transmission mechanism in this network, which can tolerate certain faults including single path delay violations. In this paper, we introduce an approach in which the end-to-end delay violation is modeled as a failure so that performance analysis can be incorporated into the overall system reliability assessment. Moreover, the well-known Fault Tree Analysis technique is employed to perform reliability assessment while taking into account the failure due to delay violations. Stochastic Network Calculus is applied to compute the upper bounds with various probability limits. A case study is carried out and the results confirm that the overall system reliability requirement can be met with less pessimistic probabilistic performance constraints.

Index Terms—AFDX, Performance Analysis, Reliability Assessment, Fault-Tree Analysis, Stochastic Network Calculus.

5.1 Introduction

Avionics Full-Duplex Switched Ethernet (AFDX) is an emerging data communication technology widely adopted by the aerospace industry for the new generation of avionic sys-

tems [54, 20, 115]. However, although industrial experiments show that the AFDX network can provide the means for high performance data communications in a wide range of avionic systems, there are still doubts about its applicability to safety-critical applications that require an extremely high level of reliability. One of the main reasons is that being an asynchronous protocol, it is difficult to accurately control the jitter introduced by multiplexing present at different levels of End Systems (ESs) and switches in AFDX networks. Therefore, it is a challenging issue to formally assess its determinism. Moreover, the AFDX standard requires that any design must ensure a guaranteed service providing a firm, mathematically provable, upper bound on end-to-end frame transit delay (see Section 3.1 in ARINC 664-P7) [13]. To meet this stringent requirement, a considerable effort in AFDX network design is devoted to end-to-end delay analysis [24, 65, 119, 66, 82]. Deterministic analysis has been applied to offer the guaranteed upper bounds on the worst-case scenarios with null occurrence probability. However, neglecting the probabilistic nature within the network often leads to over-pessimistic estimation of delay upper bounds. It is observed that the worst-case delay upper bounds obtained by using analytical tools such as Network Calculus (NC) [40, 41] are very conservative compared to the values of experimental measurements that are only about 10%-25% of the estimated ones. In contrast to deterministic analysis, stochastic approaches [47, 112, 123, 46, 128, 44] may offer more realistic performance estimation by capturing the probabilistic nature of system behavior. It is found in recent research that applying probabilistic analysis tools, such as Stochastic Network Calculus (SNC) [109], may result in a much tighter delay estimation compared to the worst-case upper bounds obtained by deterministic approaches. An immediate benefit of probabilistic analysis is that it allows relaxing the constraints in schedulability assessment and consequently may increase the network utility. However, it is still unclear on how to specify the performance requirement in terms of delay violation probabilities and how to evaluate its impact on the overall system reliability.

As a continuous effort of the recent development aimed at enhancing the determinism of frame transmission in AFDX networks [83], we propose in the present work to model the end-to-end frame transit delay violation as a failure so that performance analysis can be incorporated into the overall system reliability assessment. To this aim, SNC is applied to capture the probabilistic nature of data transmission. Moreover, the impact of the probabilistic delay bounds on the overall system reliability is evaluated using Fault Tree Analysis (FTA) analysis. Compared to deterministic approaches, the proposed idea provides both qualitative and quantitative guarantees for the safety of the overall system and may relax delay bounds with prescribed probabilities. Note that since the introduction of delay violations may degrade the reliability assessment result, the trade-off between reliability and performance should be

considered. It is worth mentioning that as the redundant data transmission mechanism is taken into account in our research, tighter bounds for performance evaluation can be expected and a better trade-off can be achieved in validating the applicability of this very promising technology to highly safety-critical avionic systems.

The expected contributions of this paper are twofold:

— introducing an approach to incorporate performance analysis into reliability assessment by considering the delay violation as a failure and establishing a corresponding reliability assessment model;

— providing a means for specifying the performance requirements based on tighter bounds associated with verified probability budgets in order to explore the fault tolerance capabilities of redundant mechanisms.

The remaining of this paper is organized as follows. Section 5.2 introduces the context of AFDX networks. Section 5.3 presents the possible failure in performance certification and the reliability analysis considering the delay violation in AFDX networks. Section 5.4 summarizes a taxonomy of end-to-end delay in the AFDX and frame transit jitter estimation using deterministic NC and SNC. A detailed case study is carried out in Section 5.5 to validate the proposed approach. Finally, some concluding remarks and directions for future research are provided in Section 5.6.

5.2 The Context of AFDX Networks

AFDX is a redundant, deterministic, full duplex and switched Ethernet technology applied in avionics communications. As shown in Figure 6.1, an AFDX network is typically composed of three types of elements: ESs, the Switches and the physical links. Each ES is connected to the switches via redundant physical links, denoted by Network A and Network B, aimed at improving the network reliability. Full duplex physical links are adopted to eliminate transmission collisions, which helps to ensure deterministic timing performance. In addition, a star topology is applied in switch connections, which makes the network scalable. Usually, it is supposed that the switch has the capability of handling parallel processing. Hence, the packets forwarded to different output ports in a switch do not interfere.

A concept that helps make AFDX deterministic is that of Virtual Link (VL) (e.g., VL1 and VL2 in Figure 6.1), which defines a logical unidirectional connection from one source ES to one or more destination ESs. Note that in AFDX networks only one ES can be the source of a VL. Every VL is labeled by a unique 16-bit identifier, ranging from 0 to 65535. Besides, in order to provide a consistent performance guarantee for VLs, the routing is statically defined offline. Furthermore, the maximum bandwidth allocated to a VL is reserved by its

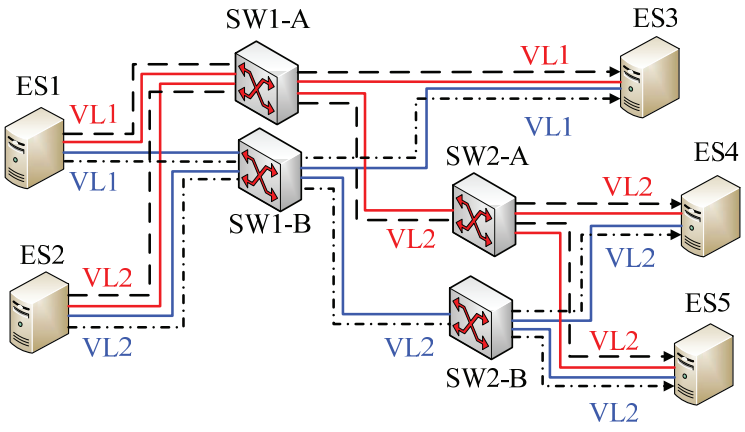


Figure 5.1 An example of AFDX network architecture.

maximum frame size (MFS) and a so-called Bandwidth Allocation Gap (BAG). According to the ARINC 664 standard, the MFS should be in the range of 64 to 1518 bytes. The BAG is the minimum time interval between successive frames in a VL (measured at start time) and should be a power of 2 multiplied by 1 ms within the set {1, 2, 4, 8, 16, 32, 64, 128}(ms). In addition, mechanisms such as frame insertion can also be considered to further enhance the determinism of AFDX networks [83].

The purpose of all these mechanisms applied in AFDX networks is to minimize the occurrence of failures to an acceptable level and as stated in the ARINC 664 standard to provide *a mathematically provable, upper bound on end-to-end frame transit delay* [13]. However, there still exists potential failures for timing certification. In the next section, we detailed an approach to handle this issue by incorporating performance analysis into reliability assessment, which allows tolerating delay violations with a prescribed probability.

5.3 Reliability Analysis with Delay Violation Probability in AFDX Networks

The main purpose of the reliability analysis is to identify the possible sources of failure and to evaluate their potential impact by using techniques such as FTA [127, 90, 27]. This section presents first the possible failures for reliability assessment in AFDX networks. Then an approach that allows incorporating performance analysis into reliability assessment is detailed and a model for FTA including the failure probability of delay violations is established.

5.3.1 Failure in AFDX Network Certification

The AFDX protocol aims at providing a high performance communication network for critical applications in an aircraft. Therefore performance certification of properties such as

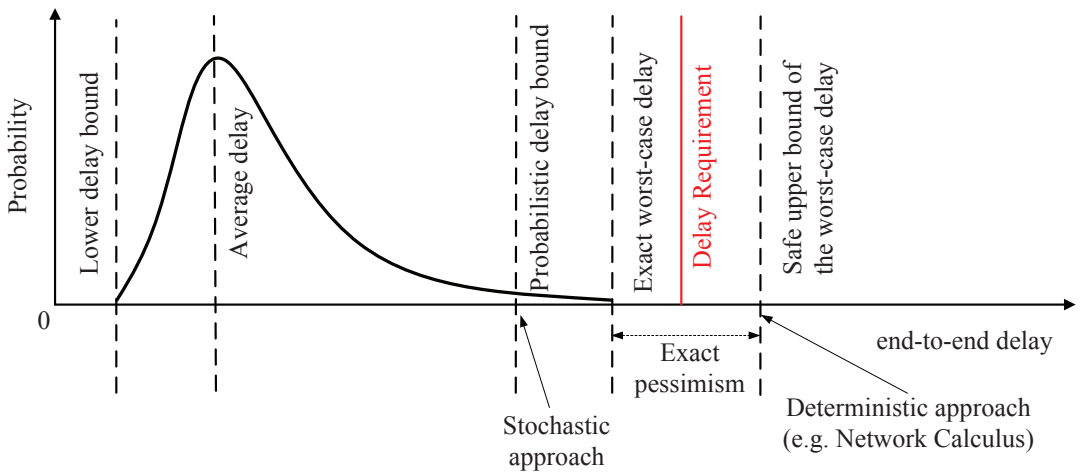


Figure 5.2 Illustration of different delay and delay upper bound definitions [20].

timing, is required regarding the generated networks and the corresponding configurations. Among the methods dedicated to timing verification, deterministic NC is a well-known, well-developed, and widely accepted technique, with which worst-case delay upper bounds can be obtained. However, the delay bounds obtained with deterministic NC are known to be pessimistic and may not meet the delay requirements for delay-sensitive functions as described in Figure 5.2. Therefore, although the worst-case upper bound may not be reached in practice, delay violations predicted by those bounds are considered as failures from the certification viewpoint. Delay violations can be characterized as recurring failures, with a similar nature as the failures induced for instance by electromagnetic interference.

In addition, deterministic analysis ignores the capability of redundant data transmission in AFDX networks, which can tolerate certain faults including delay violations on a single path. An alternative is to use SNC, which can produce tighter and more realistic probabilistic delay bounds. Although probabilistic delay bounds are attractive, there is still a need for evaluating their impact on the overall system reliability while ensuring its safety level. In the following, we present an analysis method that incorporates timing violations into a reliability assessment. In the proposed method, delay violations are considered as a type of recurring failures and it allows exploring the capability of redundant networks to tolerate them.

5.3.2 Reliability Analysis Modeling for AFDX Networks

Reliability analysis is required to conduct the safety assessment ensuring safe services during the development of a civil aircraft. Normally, the analysis is performed at three levels: aircraft level, system level, and component level. Among the approaches employed in reliability analysis, FTA is a powerful, well-developed, and widely applied technique.

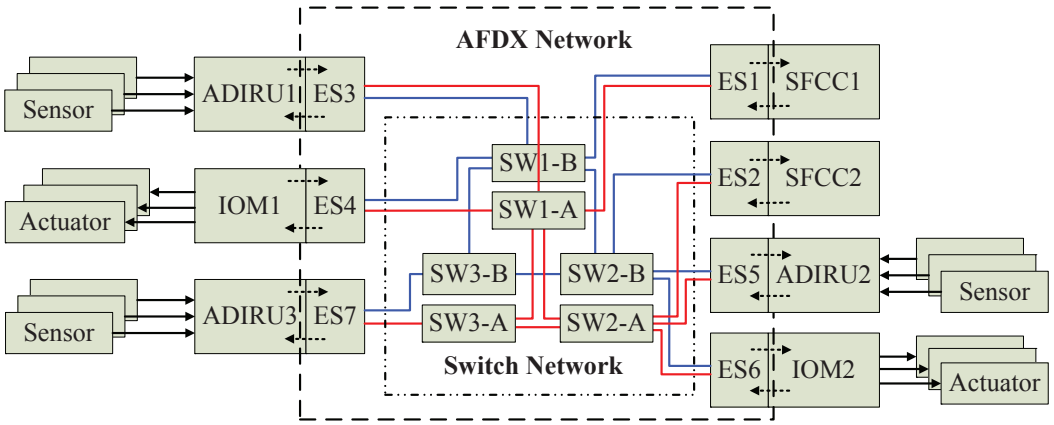


Figure 5.3 Schematic diagram of a subsystem in the redundant SFCS.

As a complete reliability analysis considering all parts of an aircraft at the same time is too complex, this paper focuses on a system involving a redundant AFDX network for a slat flap control system (SFCS). Slats and flaps installed on the leading and trailing edges of the airfoils are auxiliary control surfaces employed in low speed conditions, e.g., in takeoff and landing phases, to increase wing lift and decrease stall speed [95].

In order to guarantee the functionality of SFCS in case of failures, redundant mechanisms are employed in system design as shown in Figure 5.3, e.g., duplicating the slat and flap control computer (SFCC) and triplicating the air data inertial reference unit (ADIRU).

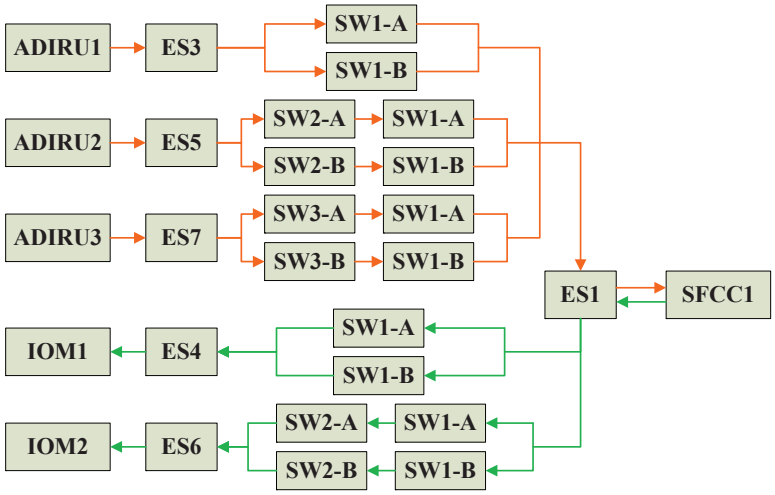


Figure 5.4 The SFCS data flow diagram in one redundant branch managed by SFCC1.

The redundant components are supposed to be functionally identical while having different hardware and software implementations to avoid the event that a fault ever affects all the redundant components. The SFCS data flow diagram in one redundant branch managed by SFCC1 is shown in Figure 5.4. More specifically, a set of sensors managed by ADIRU1, or

ADIRU2, or ADIRU3 constantly transmits information to update the status of the aircraft through the AFDX network and then, the SFCC operates as a “brain” that collects all the measured data, e.g., the speed and the position, for decision making and releases instructions across the AFDX network to IOM1 and IOM2 to control the actuators accordingly.

In the following, a fault tree is derived by tracing the possible failure sources of the predefined top event. As a fault tree comprises various parallel and sequential combinations of failures resulting in a particular undesired event, it represents a logical model that describes the relationship of the basic events leading to the predefined top event [37]. The two most common logic gates applied to show the relationship of failure effects are the AND-gate and the OR-gate [15].

As the SFCS is redundant, the loss of the whole system happens only when the two subsystems fail simultaneously. In more detail, the logic grouping of possible failure elements including delay violation failure has been constructed by considering all possible combinations as shown in Figure 5.5 [37, 15, 96, 113]. Note that as two branches of the redundant system have similar structure and data flow diagram, only one subsystem is expanded in detail. In fact, delay violation failure probability varies with the network configuration, as well as the trade-off between probability budgets and delay requirements. We provide below a brief explanation on how to perform the delay violation analysis.

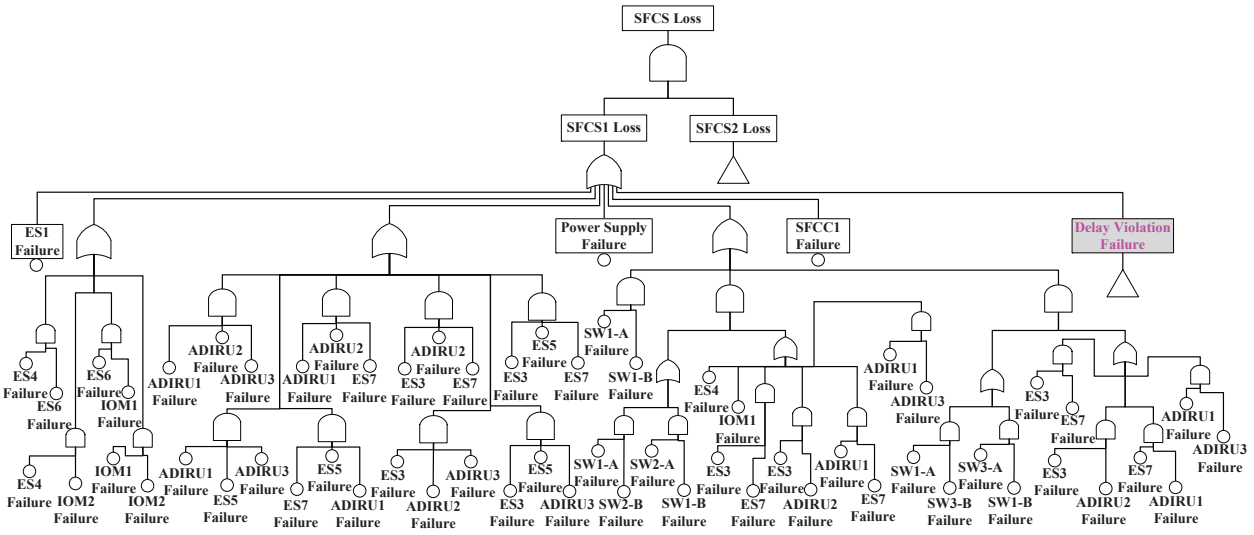


Figure 5.5 SFCS architecture Fault Tree Analysis.

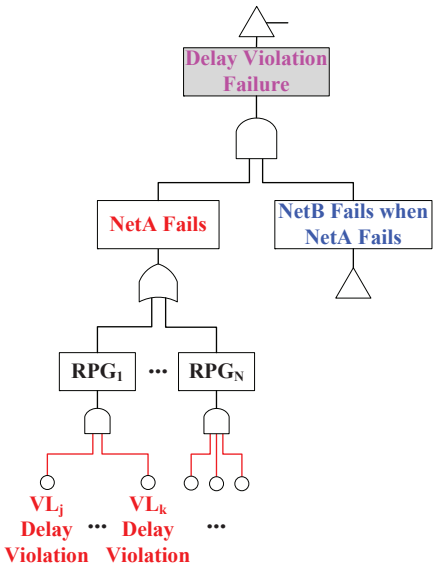


Figure 5.6 The fault tree for delay violation of VLs.

Delay violation happens when functional data required by a component of the subsystem are delivered with a delay larger than its constraint. For safety purposes, functional data may be produced by several components (ADIRU1, ADIRU2,...) and are delivered through multiple VLs in the network. Then, the set of all paths of VLs delivering directly the related data elements is called a redundant paths group (RPG) of VLs. Furthermore, all the VLs in each RPG are transmitted over two independent networks, namely Network A and Network B. Therefore, a subsystem is composed of a set of N RPGs traversing Network A and Network B. Thus it can be described as $S = \{R_1, \ldots, R_N\}$ with $R_i = \{p^i_{VL_j}\}$, $i \in [1, N]$, where $p^i_{VL_j}$ is a path of VL_j belonging to RPG R_i.

Let $D^{req}_{VL_j}$ be the delay constraint of VL_j along its paths and $D^i_{VL_j}$ be the delay of VL_j along path $p^i_{VL_j}$. The probability for $D^i_{VL_j}$ to exceed $D^{req}_{VL_j}$ is then given by $\Pr(D^i_{VL_j} > D^{req}_{VL_j})$. Since all the VLs are independent at their ingress point [109] and the delay violation distributions of the independent flows can be obtained using SNC, the probability of failure of RPG R_i is:

$$\Pr(R_i) = \bigcap_j \Pr\left(D^i_{VL_j} > D^{req}_{VL_j}\right). \tag{5.1}$$

Therefore, for Network A to fail, there needs to be at least one failure among the RPGs. Thus, the probability of failure of Network A is:

$$\Pr(NetA) = \bigcup_{i=1}^{N} \Pr(R_i) = \bigcup_{i=1}^{N} \left(\bigcap_j \Pr\left(D^i_{VL_j} > D^{req}_{VL_j}\right) \right). \tag{5.2}$$

Let $\Pr(NetB|NetA)$ be the failure probability of Network B when Network A fails. For the simplicity of analysis, we suppose that $\Pr(NetA) = \Pr(NetB)$. This implies that $\Pr(NetB|NetA) = \Pr(NetA|NetB) = \varepsilon$. Then the top level probability of delay violation failure, $\Pr_{vio}$, can be presented as:

$$\begin{aligned} \Pr_{vio} &= \Pr\left(NetA \bigcap NetB\right) \\ &= \Pr(NetB|NetA)\Pr(NetA) \\ &= \varepsilon \times \bigcup_{i=1}^{N}\left(\bigcap_{j}\Pr\left(D^{i}_{VL_j} > D^{req}_{VL_j}\right)\right), \end{aligned} \tag{5.3}$$

which will be utilized in the system fault tree analysis as shown in Figure 5.5.

Note that, being an asynchronous protocol, there is no global clock within an AFDX network. Although the redundant networks have the same workload of traffic flows, a slight difference of arrival/delivery time can incur a significantly different networking resource competition scenario and inter-flow packet forwarding routine on the switches, and this difference will have a tendency to increase on the next hop through the transmission path. Thus, the processing order of frames in the redundant networks is expected to be quite different, which leads to different jitters for the redundant frames. Furthermore, although the redundant switches execute the same functions, they utilize dissimilar hardware and software implementations [114, 132], which further randomize the process order in switches. Moreover, the frames are sent to the redundant physical channels sequentially. This further introduces unpredictable variances among the two redundant networks, as the arrival time of the frames and their copies to the redundant switches are different. Finally, the stochastic network calculus is also based on the worst case. Thus, any slight difference of processing order may lead to big jitter difference after passing through several switches. By considering all the above facts, it is reasonable to assume that, regarding the delay violation, the probability that two redundant networks fail simultaneously is significantly lower than the failure probability of a single network.

In practice, delay violation probability targets for each path can be directly assigned with respect to the reliability budgets instead of providing $D^{req}_{VL_j}$, which may be based on experience. In such a case, a stochastic upper bound can be obtained according to the probability budget and then set as $D^{req}_{VL_j}$, which can help to specify delay requirements with tighter delay bounds by considering the trade-off between reliability and performance.

Thus, performance analysis is incorporated into reliability assessment by considering possible delay violations. Furthermore, SNC is employed to obtain the delay violation probabilities

and the corresponding stochastic upper bounds, which is introduced in the following section.

5.4 End-To-End Delay Analysis in AFDX Networks

5.4.1 End-To-End Delays in AFDX Networks

In AFDX networks, data transmission delays depend on both the hardware performance and the VL scheduling (or the configuration). In more detail, the end-to-end delay of VL_i traversing N switches can be expressed as:

$$\begin{aligned} D_{VL_i} &= D_T^{Link} + \sum_{j=1}^{N} D_P^{S_j} + J_Q^i \\ &= D_T^{Link} + \sum_{j=1}^{N} D_P^{S_j} + J_Q^{ES} + \sum_{j=1}^{N} J_Q^{S_j}, \end{aligned} \tag{5.4}$$

where J_Q^i is the queueing jitter associated with VL_i and detailed definitions for other parameters are specified below.

— D_T^{Link} is the transmission latency over physical links, which is determined by the frame length, the link bandwidth, and the number of physical links. As AFDX is a full duplex switched network, there is no collision between the reception and the transmission on physical links. Suppose that all the physical links have the same transmission rate C and that the VL has a MFS of $L_{\max}$. Hence

$$D_T^{Link} \leq (N+1) \times \frac{L_{\max}}{C}. \tag{5.5}$$

— $D_P^{S_j}$ is the technological latency within the j^{th} switch, which depends on the hardware performance. According to the ARINC 664 standard, the technological latency within a switch has to be less than 100μs. Therefore, $D_P^{S_j} \leq 100\mu s$.

— J_Q^{ES} is the queueing jitter at the ES due to the congestion with other VLs. This jitter depends highly on the load at the output port when it arrives and its priority if priority-based scheduling is employed. Therefore, it is a variable value. Based on the theory of NC, the upper bound for this jitter can be computed in a deterministic or probabilistic manner.

— $J_Q^{S_j}$ is the queueing jitter at the output buffer in Switch j due to the backlog. Similar to J_Q^{ES}, this jitter depends also on output load and VL's priority.

Furthermore, D_{VL_i} can be divided into two parts: fixed latencies, D_T^{Link} and $D_P^{S_j}$, and variable jitters, J_Q^{ES} and $J_Q^{S_j}$. The fixed latency can be statically computed because they depend only

on hardware performance, such as switching processing speed, physical link bandwidth, and configuration parameters, e.g., the MFS of VLs. However, besides the factors mentioned above, the computation of variable jitters must consider the effect of other related VLs and the corresponding scheduling algorithms. In this paper, we mainly focus on jitter computation under a first-come, first-served (FCFS) policy, in which all the VLs have the same priority.

5.4.2 Deterministic Network Calculus

Deterministic NC has been successfully applied in AFDX networks to derive guaranteed delay upper bounds of individual VLs. In the computation, the service curve for each VL is used to present the resource allocation. Let $\mathcal{I}$ be the set of frames of all VLs sharing the same output port in an ES or in a switch. As the FCFS policy is used in this work, each VL may be blocked by other VLs belonging to $\mathcal{I}$ in the worst case.

Denote by $\alpha_i(t)$ the arrival curve of the VL_i in the source ES. Let $L_{\max_i}$ be the MFS of VL_i and BAG_i be the minimum frame interval of VL_i. Then the arrival curve $\alpha_i(t)$ can be expressed by

$$\alpha_i(t) = \rho_i t + \sigma_i, \tag{5.6}$$

where $\sigma_i = L_{\max_i}$ and $\rho_i = L_{\max_i}/\text{BAG}_i$.

The service curve corresponding to the VL_i is obtained as follows using deterministic NC [24]:

$$\begin{aligned}\beta_i(t) &= C_i[t - T_i]^+ \\ &= \left(C - \sum_{k\neq i, k\in\mathcal{I}} \rho_k\right)\left[t - \frac{\sum_{k\neq i,k\in\mathcal{I}} \sigma_k}{C - \sum_{k\neq i,k\in\mathcal{I}} \rho_k}\right]^+ .\end{aligned} \tag{5.7}$$

where $[t]^+$ is defined by $\max(t, 0)$. C_i and T_i are the minimum service rate and the worst-cast latency of VL_i, respectively. Then the deterministic worst-case jitter J_Q^i is bounded by the maximal horizontal difference between the arrival curve $\alpha_i(t)$ and service curve $\beta_i(t)$ given by:

$$\begin{aligned}J_Q^i &= \sup_{t\geq 0}\left(\inf_{\tau\geq 0}\{\alpha_i(t) \leq \beta_i(t+\tau)\}\right) \\ &= T_i + \frac{\sigma_i}{C_i} = \frac{\sum_{k\in\mathcal{I}} \sigma_k}{C - \sum_{k\neq i,k\in\mathcal{I}} \rho_k}.\end{aligned} \tag{5.8}$$

Based on the principle of "pay bursts only once" [78], the service curve for VL_i passing cascaded switches can be concatenated by using the convolution under the min-plus algebra.

The min-plus convolution is defined as follows:

$$(\beta_i \otimes \beta_k)(t) = \inf_{0 \le \tau \le t} \{\beta_i(\tau) + \beta_k(t-\tau)\}. \tag{5.9}$$

By using the equivalent model, a tighter bound is obtained, since bursts only delay transmission once in the analysis.

5.4.3 Stochastic Network Calculus

In contrast to deterministic NC mentioned above, SNC can offer a distribution of delay upper bounds with a prescribed probability. The probabilistic approach is well-suited for redundant AFDX networks because AFDX provides dual routes for every VL, which allows VLs to have the capability to tolerate a delay violation. Therefore, SNC can be used to provide more realistic and tighter delay bounds compared to the deterministic ones.

Suppose that we have an aggregation of heterogeneous flows, $\mathcal{I}$, that are dispatched from the same output port. Then the aggregated arrival curve for the set $\mathcal{I}$ can be expressed as $\alpha(t) = \sum_{i=1}^{I} \alpha_i(t)$, for $i \in \mathcal{I}$. For the rate-latency service curve $\beta(t) = C[t-e]^+, e \ge 0$. Denote by $Q(t)$ the backlog of the aggregated set $\mathcal{I}$ at time t. Let further τ be the intersection of the arrival curve $\alpha(t)$ and the service curve $\beta(t)$, hence

$$\tau = \inf\{t \ge 0 | \alpha(t) \le \beta(t)\} = \frac{C \times e + \sum_{i=1}^{\mathcal{I}} \sigma_i}{C - \sum_{i=1}^{\mathcal{I}} \rho_i}. \tag{5.10}$$

Then the probability that $Q(t)$ exceeds a given number q can be bounded by [125]:

$$\Pr(Q(t) > q) \le \sum_{k=0}^{K-1} \exp(-g(s_k, s_{k+1})), \tag{5.11}$$

where $K \in \mathbb{N}$, and $s_k = k\tau/K$ $(k = 0, \ldots, K)$. The function $g(u, v)$ is defined as:

$$g(u,v) = \frac{2\left([q + \beta(u) - \rho v]^+\right)^2}{\sum_{i=1}^{I} \alpha_i(v)^2}. \tag{5.12}$$

Note that the parameter K is set to the value that produces the minimum probabilistic upper bounds in this paper, although (5.11) holds for any $K \in \mathbb{N}$.

We further apply another result proved in [124], with which the probability distribution of

jitter upper bounds for the flows can be expressed as:

$$\Pr(J_Q^i > t) \leq \frac{C}{\rho} \Pr\left(Q(t) > Ct\right). \tag{5.13}$$

The principle of "pay bursts only once" applies also to the SNC when the VLs traverse multiple switches. The equivalent model can be employed in the stochastic calculation to obtain tighter bounds. A case study is carried out in the following section to illustrate delay bounds computation and its impact on reliability analysis.

5.5 Case Study and Evaluation Results

In this section, we take the aforementioned SFCS as an example to perform a case study, in which AFDX is employed in the communication network.

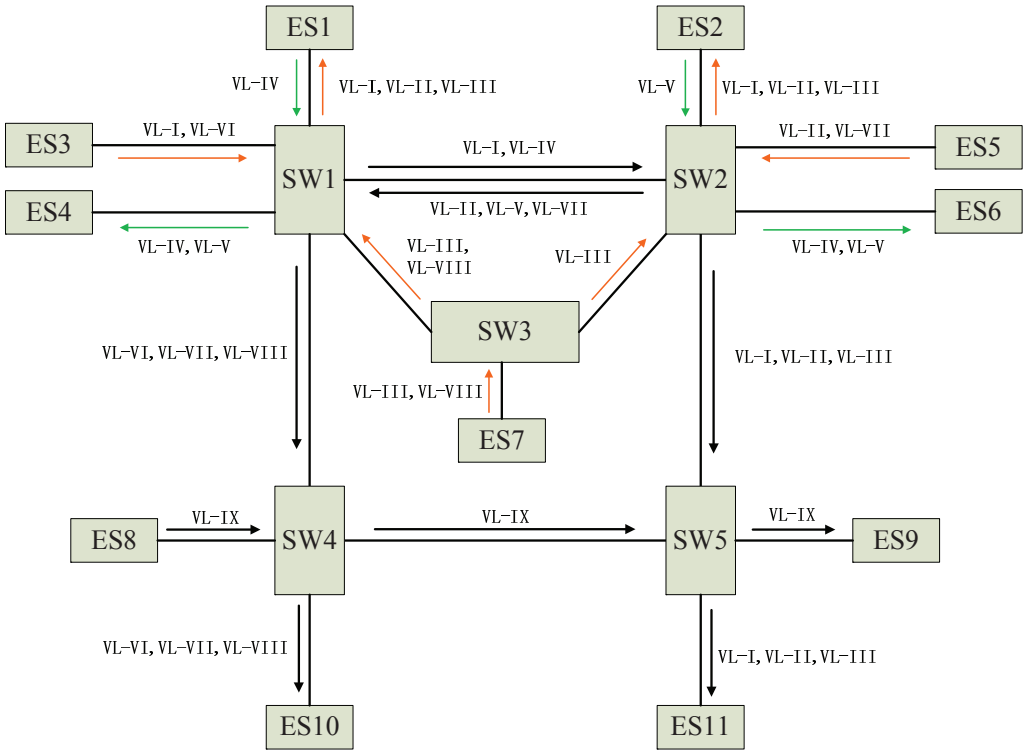


Figure 5.7 An AFDX network for communications within SFCS.

The network structure is shown in Figure 5.7. The whole network is composed of 5 switches, 11 ESs, 140 VLs and 300 VL paths. The network configuration is specified in Table 5.1, in which aggregations having the same functionality can be grouped as follows: {VL-I, VL-II,

VL-III}, {VL-IV, VL-V}, {VL-VI, VL-VII, VL-VIII}. Note that the analysis method does not depend on the complexity of the network.

Table 5.1 AFDX network configuration

VL Aggregate		BAG (ms)	$L_{\max}$ (byte)	VL Number	Hop Number
VL-I	Path1 (ES3→SW1→ES1)	2	160	20	1
	Path2 (ES3→SW1→SW2→ES2)				2
	Path3 (ES3→SW1→SW2→SW5→ES11)				3
VL-II	Path1 (ES5→SW2→ES2)	2	160	20	1
	Path2 (ES5→SW2→SW1→ES1)				2
	Path3 (ES5→SW2→SW5→ES11)				2
VL-III	Path1 (ES7→SW3→SW1→ES1)	2	160	20	2
	Path2 (ES7→SW3→SW2→ES2)				2
	Path3 (ES7→SW3→SW2→SW5→ES11)				3
VL-IV	Path1 (ES1→SW1→ES4)	8	240	20	1
	Path2 (ES1→SW1→SW2→ES6)				2
VL-V	Path1 (ES2→SW2→ES6)	8	240	20	1
	Path2 (ES2→SW2→SW1→ES4)				2
VL-VI		4	180	10	2
VL-VII		4	180	10	3
VL-VIII		4	180	10	3
VL-IX		64	300	10	2

The communication network of the SFCS involves ES1-ES7 and SW1-SW3. As listed in Table 5.1, the VLs employed by the SFCS traverse one or more switches. Each VL may be

delayed either in the source ESs or in switches. For example, VL-I is directly influenced by VL-VI and VLs in its aggregate at the output port of ES3. Although VL-VI is not utilized by SFCS, it should be considered when performing the delay analysis of SFCS as VL-VI contributes to the delay of VL-I in the source ES. Note that the VLs forwarded to different output ports of one switch have no influence on each other. Consequently, VL-I may be delayed by VL-II and VL-III in SW1, where VL-IV, VL-V, VL-VII and VL-VIII have no influence on VL-I.

As presented in Section 5.4, the end-to-end delay experienced by one VL can be divided into the fixed latency and the variable jitter. Obviously, VL delay violations may be caused by jitter due to possible VL conflicts. Hence

$$\Pr\left(D_{VL_i} > D^i_{UB}\right) = \Pr\left(J^i_Q > J^i_{UB}\right),$$

where J^i_{UB} represents the expected jitter upper bound for VL_i.

In this work, we assume that each physical link offers a constant rate C=100M bits/s and the variable jitter of one VL cannot exceed its BAG. Then, we can calculate the worst-case jitter bound for each VL using (5.8). Moreover, the distribution of the probabilistic jitter bounds can be given based on (5.13). Using the same approach, we can obtain a probabilistic distribution for each aggregate. As an example, the probabilistic jitter bound distribution and deterministic jitter upper bound for VL-I (Path2) are depicted in Figure 5.8.

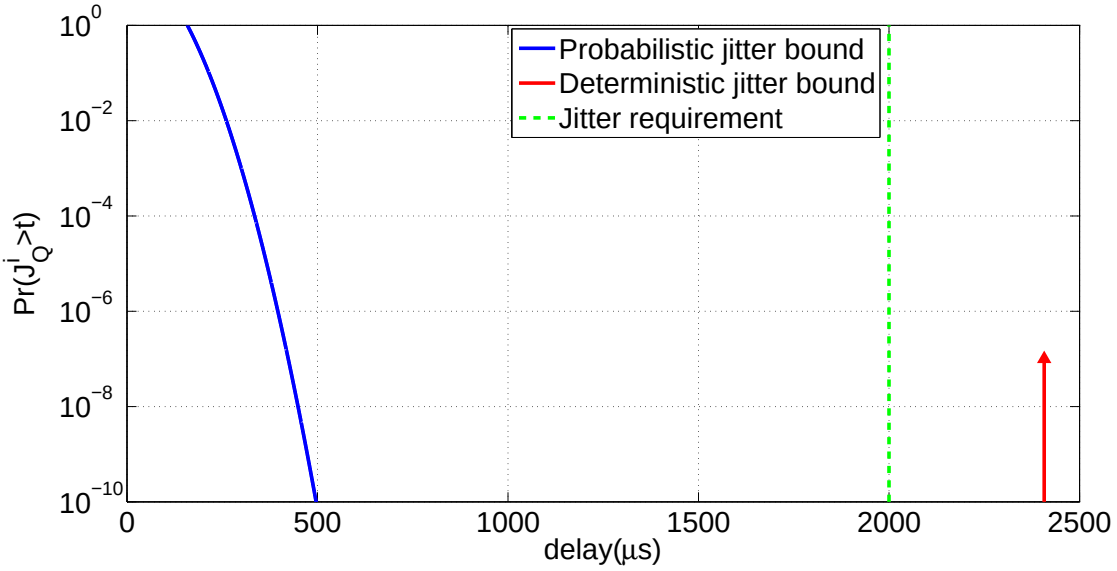


Figure 5.8 Distribution of probabilistic jitter bound and deterministic jitter upper bound for VL-I (Path2).

Table 5.2 Results obtained with network calculus

VL Aggregate		BAG (μs)	Deterministic Jitter Bound (μs)	Probabilistic Jitter Bound (μs)	
				10^{-9}	10^{-6}
VL-I	Path1 (ES3→SW1→ES1)	2,000	1,555	372	311
	Path2 (ES3→SW1→SW2→ES2)		2,407	473	396
VL-II	Path1 (ES5→SW2→ES2)	2,000	1,555	372	311
	Path2 (ES5→SW2→SW1→ES1)		2,407	473	396
VL-III	Path1 (ES7→SW3→SW1→ES1)	2,000	1,555	372	311
	Path2 (ES7→SW3→SW2→ES2)		1,555	372	311
VL-IV	Path1 (ES1→SW1→ES4)	8,000	847	417	350
	Path2 (ES1→SW1→SW2→ES6)		1,315	456	382
VL-V	Path1 (ES2→SW2→ES6)	8,000	847	417	350
	Path2 (ES2→SW2→SW1→ES4)		1,573	482	403

As highlighted in Table 5.2, VL-I (Path2) and VL-II (Path2) under worst-case performance evaluation cannot meet the requirement when the upper bound for the jitter is limited to the BAG. Instead of rescheduling or changing the scheduling policy, SNC can be applied to find probabilistic jitter bounds, which not only satisfy the BAG limitation but also provide tighter bounds for performance evaluation. The admissible probabilistic upper bounds for each VL aggregate were obtained assuming the assigned probability budgets are respectively set to 10^{-9} and 10^{-6} as listed in Table 5.2. Next, we checked whether the probabilistic delay bounds are safe for the SFCS.

Suppose that the reliability for the SFCS should meet catastrophic failure requirement, that is 10^{-9} per flight hour. This can be combined with information related to flight duration.

Indeed, according to statistics from Airbus in September, 2012, the A380s have accumulated over 600,000 flight hours in more than 72,000 flights [1]. Thus the average flight duration is about 8.3 hours. Another factor to consider is the utilization time of the SFCS in each flight. Suppose further that, in normal operation, an aircraft flies at low speed during about 10 minutes per flight on average, including takeoff, ascent, descent and landing. Therefore, the hourly failure rate requirement for SFCS, λ^R_{SFCS}, is

$$\begin{aligned}\lambda^R_{SFCS} &= \frac{10^{-9}}{\text{flight hour}} \times \frac{8.3\text{flight hours}}{\text{flight}} \times \frac{1 \text{ flight}}{10 \text{ minutes}} \times \frac{60 \text{ minutes}}{\text{hour}} \\ &\approx 5.0 \times 10^{-8}/\text{hour},\end{aligned}$$

which means that the failure rate of the top event has to be less than 5.0×10^{-8}/hour.

It is further assumed in this paper that failure probabilities are directly related to component exposure times T, then the failure rate λ can be approximated as:

$$\lambda \approx \frac{\Pr(x)}{T}. \tag{5.14}$$

Therefore, when a failure probability budget of 10^{-6} is selected for every VL to obtain a tighter upper bound, we have the failure rate for each VL based on (5.14): $\lambda_{VL_i} \approx 1.2 \times 10^{-5}$/hour, where $T = 8.3$ hours. In this analysis, all the VLs are assumed to be grouped into RPGs based on their functionalities. Then following our earlier discussion about the fault tree for delay violation of VLs as shown in Figure 5.6, the calculation can be performed with (5.3) and (5.14). The computed delay violation failure rate in Network A is about 2.9×10^{-9}/hour. Suppose that $\varepsilon = 10^{-3}$, which is indeed a very conservative estimation of the conditional probability $\Pr(NetB|NetA)$ (or $\Pr(NetA|NetB)$). Then the delay violation failure rate in one subsystem is about 2.9×10^{-12}/hour.

Table 5.3 List of component failure rate

Components	Failure Rate λ (/hour)
SFCC1, SFCC2	10^{-7}
ADIRU1, ADIRU2, ADIRU3	10^{-5}
ES1, ES2, ES3, ES4, ES5, ES6, ES7	10^{-5}
SW1-A, SW1-B, SW2-A, SW2-B, SW3-A, SW3-B	10^{-5}
IOM1, IOM2	10^{-5}

The failure rate budgets for other basic components are given based on experience and safety

requirements. Accordingly, the failure rate of SFCC is assigned at the level of 10^{-7}/hour [47] and the other components are classified into the major failure condition, which is associated with a failure rate of 10^{-5}/hour [15] (as shown in Table 5.3). Following our earlier discussions about the fault tree of SFCS (see Figure 5.5), the calculation is executed using OpenFTA [102] and we obtain:

$$\tilde{\lambda}^{C}_{SFCS} \approx 5.0 \times 10^{-10}/\text{hour} \ll \lambda^{R}_{SFCS}.$$

The contribution of delay violation to the top event is about 6.0×10^{-3}. In the given fault tree, the selected delay violation probability budgets of VLs can then be certified. Finally, it is worth noting that the results given in Table 5.2 show that probabilistic jitter bounds for all the VLs are reduced by more than 50% compared with the worst-case upper bounds in the considered case study.

5.6 Conclusion

This paper presented an approach to incorporate the performance analysis into a quantitative reliability assessment, which may allow for the adoption of probabilistic upper bounds in AFDX network certification. The reliability analysis of AFDX networks is performed with the investigated SFCS by using the FTA technique, in which we also present how to integrate VL delay violation probabilities into the reliability analysis. In addition, much tighter probabilistic upper bounds have been obtained using SNC. The safety of the probabilistic bounds has been demonstrated in a case study, which shows that the system reliability requirements can still be met even considering a certain probability of VL delay violations. Moreover, the results show that the probabilistic upper bounds are significantly less pessimistic than the deterministic ones, which can facilitate network design by offering a larger margin regarding delay requirements for delay-sensitive applications.

It is worth noting that for simplicity the scheduling policy considered in the present work is FCFS. Different scheduling strategies may significantly impact the jitter probability distribution. Nevertheless, the proposed approach is applicable to more generic scheduling policies and network configurations.

CHAPTER 6 ARTICLE 3: RELIABILITY ENHANCEMENT OF REDUNDANCY MANAGEMENT IN AFDX NETWORKS

In order to improve data availability, AFDX adopts a redundancy management scheme for frame transmission. However, it has been reported in the specification that this redundancy management mechanism can fail in some special cases. Therefore, this chapter proposes a mathematical analysis of the phenomenon and addresses these failures in the redundant transmission management of AFDX networks. The following sections are based on [85], which has been submitted to IEEE Transactions on Industrial Informatics.

Authors—Meng Li, Guchuan Zhu, Yvon Savaria, and Michaël Lauer.

Abstract—AFDX is a safety critical network in which a redundancy management mechanism is employed to enhance the reliability of the network. However, as stated in the ARINC 664-P7 standard, there still exists a potential problem, which may fail redundant transmissions due to sequence inversion in the redundant channels. In this paper, we explore this phenomenon and provide a mathematical analysis. It is revealed that the variable jitter and the transmission latency difference between two successive frames are the two main sources of sequence inversion. Thus, on one hand, a staircase model is applied to characterize the arrival curve in order to obtain tighter jitter bound estimations. On the other hand, two methods are proposed and investigated to mitigate the jitter pessimism, which can eliminate the potential risk. A case study is carried out and the obtained results confirm the validity and applicability of the developed approaches.

Index Terms—Reliability Enhancement, AFDX, Virtual Link, Fault Tolerance.

6.1 Introduction

Reliability is one of the main concerns for safety-critical systems (See, e.g., [98, 94, 70, 45, 133, 4]). A typical example of such systems is avionics communication network for which failures may be catastrophic. Therefore, guaranteeing a reliable communication among avionics systems at every flight phase is critical for aircrafts. To ensure that stringent reliability requirements are met, certain standards, e.g. ARINC 429, have been developed and successfully deployed since the late 1970s [53]. However, as the amount of electronic components in an aircraft continues to increase, legacy avionics communication protocols are at their limit in terms of performance and design complexity. Among the available technologies for handling the new challenges in avionics systems design, we can find an Ethernet-based technology,

namely the Avionics Full Duplex Switched Ethernet (AFDX)[18], which features high speed, low cost, high flexibility, and reduced weight because of less wiring.

Built on the basis of Ethernet technology, the AFDX not only offers a high available bandwidth and a high communication speed, but also provides deterministic performance, which is the most prominent challenge to using such a technology in avionics. In AFDX networks, determinism is enforced mainly through the concept of Virtual Link (VL), which defines a logical unidirectional connection and a bounded data transmission bandwidth. Besides, the allocated bandwidth is reserved by VL's maximum frame size (MFS) and the so-called Bandwidth Allocation Gap (BAG), which defines the minimum time interval between successive frames in a VL. Furthermore, the AFDX network is composed of two independent and redundant networks, which provides the required reliability for ensuring its determinism. Consequently, the unavoidable faults on single paths can be tolerated by the redundancy management mechanism.

Nevertheless, although the AFDX was originally developed for safety critical avionic applications, it has not yet been used in critical systems that require the highest level of reliability, e.g., flight control systems [63]. One of the main reasons is that the redundancy management mechanisms in AFDX networks may fail with a relatively high probability. Specifically, as pointed out in ARINC 664-P7 (see, Section 3.2.6 in [13]), the redundant transmission mechanism fails if the following two events occur simultaneously: (1) a frame is lost during transmission on one of the redundant networks; (2) the subsequent frame on the network with frame loss arrives earlier to the destination End Systems (ES) than the copy of the lost frame sent through the other network, which is called a sequence inversion in the redundant channels. Consequently, the copy of the lost frame is discarded by redundancy management mechanisms due to the employed "First Valid Wins" (FVW) policy. Obviously, this degrades the network reliability. In real avionics communication networks, frame loss, even if observed with a very small probability, is inevitable. Therefore in order to guarantee the reliability of the redundancy management mechanism, one must prevent the second condition from occurring. This is a real challenge to system designers, due to the lack of an analytical framework for this problem.

The motivation of this paper is to provide a mathematical analysis of redundancy management (RM) failures in the AFDX protocol. It is revealed that the sequence inversion phenomenon, which can result in the invalidity of the redundancy management mechanism, is due to the variable jitter and the transmission latency difference between two successive frames. To tackle this problem, first a staircase model for the arrival curve is applied in order to obtain tighter jitter bounds to mitigate the pessimism in jitter estimation, so that

it might be possible to assess that Condition (2) will not occur. Furthermore, two methods that can contribute to eliminate the sequence inversion problem are proposed. One of these methods is based on local synchronization (LS) [57, 86] and the other exploits the notion of transmission latency difference minimization (TLDM) proposed in this work. This allows enhancing the reliability of RM. We show that these two approaches help mitigating the delay difference between two redundant networks in the worst case. A case study is carried out and the obtained results confirm the validity and the applicability of the developed approaches.

To the best of our knowledge, this is the first work that presents a formal mathematical analysis on the sequence inversion problem. Specifically, the main contributions of this paper are:

— identifying the sources of sequence inversion and providing a mathematical analysis regarding potential failures in RM;
— applying the staircase model for the arrival curve to mitigate the pessimism in jitter estimation and obtain tighter upper bounds;
— introducing two approaches that can eliminate potential failures due to frame sequence inversion of the redundant networks.

The present work contributes to the efforts aimed at enhancing the determinism and the reliability of AFDX networks to render this promising technology applicable for mission-critical avionics systems.

The remaining of the paper is organized as follows. Section 6.2 introduces the context of AFDX networks. Section 6.3 describes some potential failures in redundant AFDX networks and provides a corresponding mathematical analysis. Section 6.4 applies a staircase model to provide a tighter jitter estimation. Then, in Section 6.5 two approaches are developed to enhance the reliability of RM. In Section 6.6, a case study is carried out to validate the developed approaches and to evaluate the obtained performance. Finally, some concluding remarks and directions for future research are provided in Section 6.7.

6.2 The Context of AFDX Networks

6.2.1 Basis of AFDX Networks

An AFDX network is typically composed of three types of elements: ESs, switches and physical links. Each ES is connected to the switches via redundant physical links, denoted by Network A (-A suffix to switches) and Network B (-B suffix to switches) as shown in Figure 6.1. Full duplex physical links are adopted to eliminate transmission collisions, which help to ensure deterministic timing performance. In addition, a star topology is applied

in switch connections, which makes the network scalable. Usually, it is supposed that the switches have the capability of handling parallel processing. Hence, there is no interference between the packets forwarded to different outputs.

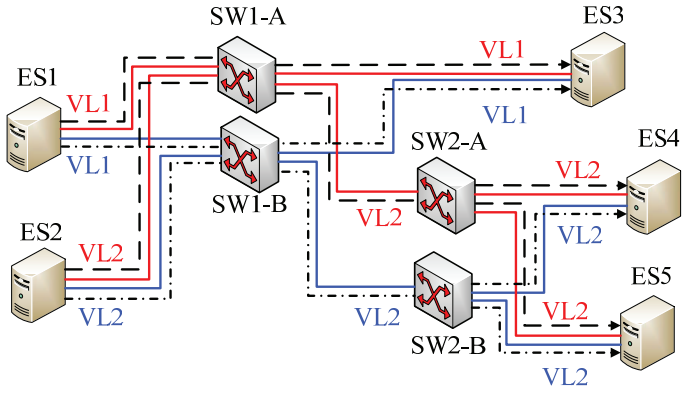


Figure 6.1 A simple AFDX network.

The concept of VL is used in AFDX to enhance the network determinism, which is the communication mechanism between ESs. Specifically, a VL defines a logical unidirectional connection from one source ES to one or more destination ESs. In AFDX networks, only one ES can be the source of a VL and the routing of VLs is statically defined off-line.

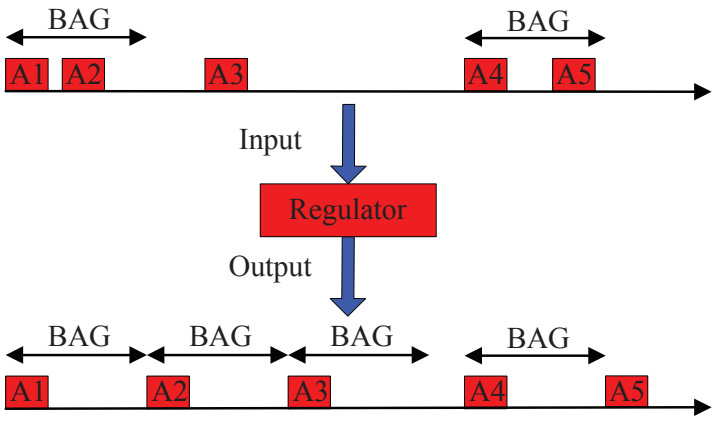


Figure 6.2 The regulation of VL flow.

As shown in Figure 6.2, input frames, either periodic or aperiodic, are regulated by the BAG, through which the instantaneous frame rate of a VL is limited. Therefore, the maximum bandwidth allocated to a VL is determined by its MFS and BAG [13]. According to the ARINC 664-P7 standard, the MFS should be in the range of 64 to 1518 Bytes, including a

header of 47 Bytes. It also needs to take into account an overhead of 20 Bytes (Interframe Gap+Preamble+Start Frame Delimiter) during frame transmission. The BAG should be a power of 2 multiplied by 1 ms within the set {1, 2, 4, 8, 16, 32, 64, 128}(ms).

Scheduling in an ES or a switch is performed on a per VL basis, which may introduce jitters due to the congestion of VLs at the outputs. In any case, the jitter introduced by multiplexing at the output of an ES is required to be bounded by 500μs [13]. Furthermore, traffic policing is applied in switches to protect the network from babbling-idiot failures. The characteristics of VLs and the traffic shaping and policing mechanisms applied in ESs and switches are essential for guaranteeing that the end-to-end delay of each frame can be upper bounded.

6.2.2 Redundancy Management

As shown in Figure 6.1, in an AFDX network, the frames in a VL are transmitted through two redundant and independent paths to achieve a high level of communication reliability. As the switches are not aware of the redundancy management mechanism, the RM is performed at the destination ES as shown in Figure 6.3.

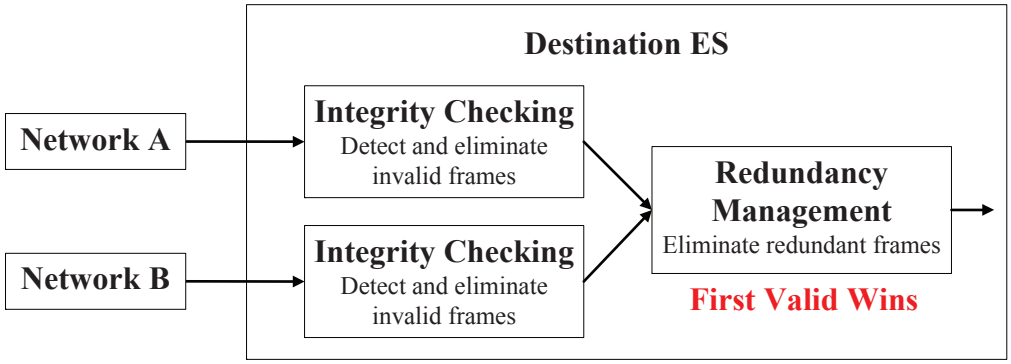


Figure 6.3 Redundancy Management in destination ES [13].

The RM is performed after integrity checking, and hence only the valid frames are processed at this stage. The basic rule used in AFDX redundancy management is the policy FVW. That is the first valid frame will be retained while the other one will be discarded. Two parameters, namely the sequence number (SN) and SkewMax, are used to identify redundant frames. Each transmitted frame is indexed by a SN ranging from 0 to 255, which is initially set to 0 and will be increased by 1 for each consecutive transmission of the same VL. The SN wraps around to 1 following the value of 255. Denote by i the index of a SN. Then the

wrap-around operation for SNs can be computed as:

$$i + 1 = (i \bmod 255) + 1. \tag{6.1}$$

Two redundant frames must have identical SNs received in an interval less than the predefined SkewMax. Otherwise, the latter reception is considered as a new frame. Hence, SkewMax is the upper bound of transmission delay difference for the redundant frames with identical SN.

Although the redundant design in AFDX networks enhances its fault tolerance, there still exist potential situations where the redundancy management mechanism may fail to manage redundant frames, which results in frame losses. In the next section, we detail this issue and provide a related mathematical analysis.

6.3 Transmission Failures in AFDX Networks

6.3.1 Frame Loss Resulting from Sequence Inversion

Although AFDX networks provide a highly reliable communication via redundant networks, the RM may fail in some special cases. Such possible failure cases have been identified in the standard ARINC 664-P7 (see, Section 3.2.6 in [13]), which may occur when a frame is lost on the faster network.

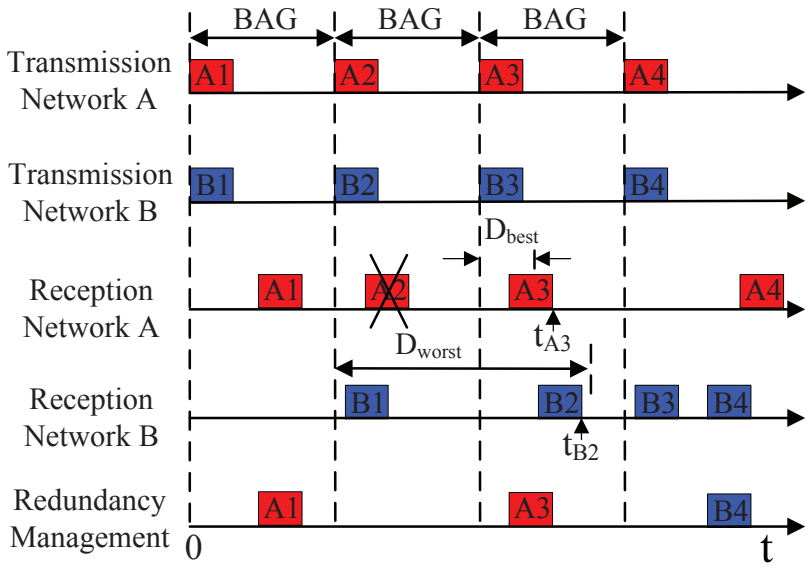


Figure 6.4 Impact of a frame lost in a redundant AFDX network due to a transmission failure on the faster network.

For example, let us consider two redundant networks, Network A and Network B, that transmit their frames every BAG interval as shown in Figure 6.4. Suppose that one frame on the faster network, e.g., A2 on Network A, is lost during transmission, e.g., due to bit errors corrupting the frame contents. To tolerate such failures, redundancy is employed in AFDX networks to increase the network reliability. However, if frame A3 arrives earlier than the frame B2 as shown in this example, a frame loss failure happens in spite of the redundant transmission. This results from the destination ES applying the FVW policy. Essentially, the lost frame in the redundant AFDX network is caused by a frame sequence reversal of Network A and Network B at the destination ES.

6.3.2 Mathematical Analysis of the Frame Sequence Inversion

In the following, we provide a detailed mathematical analysis of the frame sequence inversion phenomenon. The analysis is based on three assumptions: (1) the redundant frames are fed to the 2 redundant networks simultaneously at the source ES; (2) Network A and Network B have identical topology and configurations; (3) the technological latency in switches is considered to be a constant. Note that these assumptions are used only for the purpose of simplifying the presentation and the relaxation of these assumptions will not introduce any technical difficulty.

Denote by D_{worst} the worst-case upper delay bound experienced by the frames with maximum size in a VL. Let the transmission latency be the transmission time over the physical links. Thus D_{best}, the minimum frame delay, can be taken as the sum of technology latencies and transmission latency, which is determined by the routing of the corresponding VL and the minimum frame size. The difference between D_{worst} and D_{best} is due to the variance of frame size and the jitter caused by the influence of other VLs that share the output ports in source ES or in switches. We further assume that the redundant networks have the same topology and configuration. Then the VLs in both networks have the same parameters with respect to D_{worst} and D_{best}. For example, in the case shown in Figure 6.4, the delay of A3 cannot be smaller than D_{best} and the delay of B2 cannot exceed D_{worst}. Note that data transmission is considered to be completed when the last bit of the frame is received. Then, the reception is accomplished at t_{A3} for A3 and t_{B2} for B2, respectively. Assume that the first frame transmission starts at zero, then for the reception of A3 and B2 we have:

$$\begin{cases} t_{\text{A3}} \geq 2\text{BAG} + D_{\text{best}}, \\ t_{\text{B2}} \leq \text{BAG} + D_{\text{worst}}. \end{cases} \tag{6.2}$$

If A3 arrives earlier than B2, then we have $t_{\text{A3}} < t_{\text{B2}}$. Considering the constraints in (6.2),

we can obtain

$$D_{\text{worst}} - D_{\text{best}} > \text{BAG}. \tag{6.3}$$

Denote by $L_{\max}$ and $L_{\min}$ the maximum and minimum frame sizes of the VL, respectively. Let C be the transmission rate of the physical links and n be the number of physical links the VL traverses. In that case, we have

$$D_{\text{worst}} - D_{\text{best}} = J_{e2e} + \left(L_{\max} - L_{\min}\right)/C \times n,$$

where J_{e2e} represents the end-to-end jitter upper bound induced by its burst and other VLs during data transmission. Note that the order of frames belonging to a VL on each path is maintained by the switches to guarantee no frame sequence inversion. Therefore, order inversion can only occur in destination ES for the frames belonging to different paths.

6.3.3 Condition for Avoiding Frame Sequence Inversion

In order to avoid the possible failure due to frame sequence inversion, the transmission delay difference between any two successive frames, that have different SN and come from different paths, should be restricted within a BAG. Note that the transmission delay difference is different from the previously mentioned parameter SkewMax. Let $D_{\text{A}}(i)$ ($D_{\text{B}}(i)$) and $D_{\text{B}}(i+1)$ ($D_{\text{A}}(i+1)$) represent the delay experienced by two consecutive frames, where the index i represents a SN and $i \in [0, 255]$. Denote by $J_{\text{A}}(i)$ and $J_{\text{B}}(i)$ the jitters experienced by the frames with index i traversing Network A and Network B, respectively. Then we have:

$$D_{\text{A}}(i) - D_{\text{B}}(i+1) \leq J_{\text{A}}(i) - J_{\text{B}}(i+1) + \frac{n \times \left[L_{\text{A}}(i) - L_{\text{B}}(i+1)\right]^{+}}{C}, \tag{6.4}$$

where $[\cdot]^{+}$ is defined by $\max(\cdot, 0)$. Note that $L_{\text{A}}(i) = L_{\text{B}}(i)$ and $L_{\text{A}}(i+1) = L_{\text{B}}(i+1)$. The constraint for $(D_{\text{B}}(i) - D_{\text{A}}(i+1))$ can be obtained similarly as for (6.4). As the introduced jitter has an upper bound of J_{e2e} and a lower bound of 0, both $(J_{\text{A}}(i) - J_{\text{B}}(i+1))$ and $(J_{\text{B}}(i) - J_{\text{A}}(i+1))$ are upper bounded by J_{e2e}. Furthermore, denote by $D_{TLD}(i)$ the transmission latency difference between two consecutive frames, and then the general expression can be given as:

$$D_{TLD}(i) = \frac{n \times \left[L(i) - L(i+1)\right]^{+}}{C}. \tag{6.5}$$

Thus, the condition to avoid the possible failure is given by:

$$J_{e2e} + \max_{i}\{D_{TLD}(i)\} < \text{BAG}. \tag{6.6}$$

The first part on the left-hand side of this inequality represents the maximum jitter introduced by the VL frame with the maximum size and other VLs during transmission, and the second part denotes the maximum transmission latency difference between two successive frames.

6.4 Tightening End-To-End Delay Analysis Using A Staircase Arrival Curve

It is revealed in (6.6) that frame sequence inversion is caused by the frame size difference and the jitter in frame transmission. Thus, a firm, mathematically provable, upper bound on end-to-end frame transit is required to accurately perform the potential failure analysis discussed in Section 6.3. The challenge is raised due to the asynchronous protocol adopted by the AFDX, as it is difficult to accurately control the jitter introduced by multiplexing appearing at different levels of ESs and switches in the network. By applying suitable theoretical tools, such as network calculus [40, 41, 78, 24, 109] and the trajectory approach [19, 21, 20, 65, 22], the worst-case upper delay bounds can be obtained. However, the bounds obtained using these tools were proved to be pessimistic in general [33]. In this section, a staircase arrival curve model is introduced to find tighter delay upper bounds. More details about the staircase arrival curve can be found in [78]. Note that the following analysis is performed based on FIFO scheduling.

6.4.1 Staircase Arrival Curve Model

One of the most widely used VL data traffic models is the affine arrival curve [24, 20, 58], which can be expressed by:

$$\alpha_{\rho,\sigma}(t) = \rho t + \sigma,\ t \geq 0, \tag{6.7}$$

where σ is the burst transmission of the VL, $\sigma = L_{\max} + 20$, and $\rho = \sigma/\text{BAG}$. In (6.7) a transmission overhead of 20 bytes is grouped in the arrival curve. As the input of a VL is a sequence of packets and is regulated on a per VL basis to guarantee a minimum BAG interval between two continuous frames, stair functions can also be employed to define the arrival curve. The staircase arrival curve $\alpha_{\mathrm{T},\tau}(t)$ can be expressed by:

$$\alpha_{\mathrm{T},\tau}(t) = \left\lfloor \frac{t+\tau}{T} \right\rfloor \sigma;\ t, \tau \geq 0, \tag{6.8}$$

where σ is the burst transmission of the VL, $\sigma = L_{\max} + 20$, and $T = \text{BAG}$. Obviously,

$$\left\lfloor \frac{t+\tau}{T} \right\rfloor \sigma \leq \frac{t+\tau}{T}\sigma = \rho(t+\tau);\ t, \tau \geq 0.$$

As shown in Figure 6.5, the staircase arrival curve $\alpha_{\mathrm{T},\tau}(t)$ ($\tau = \mathrm{BAG}$) is tighter than the affine arrive curve $\alpha_{\rho,\sigma}(t)$, which may lead to more accurate and less pessimistic upper bounds.

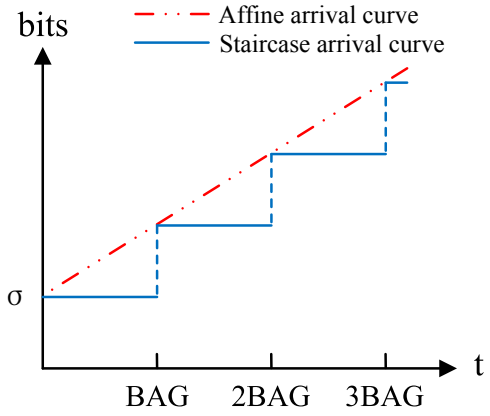


Figure 6.5 Examples of an affine arrival curve and a staircase arrival curve (τ=BAG).

6.4.2 Arrival Curve of Output Flow Under the Staircase Model

In this section, the calculation of the arrival curve for an output flow is detailed based on the staircase model. Suppose that the physical link offers a constant service rate C for a set of VLs belonging to an aggregate $\mathcal{I}$. Let $\beta(t) = Ct$ be the service curve provided by the scheduler. Let $\alpha_{\mathcal{I}}$ denote the arrival curve for the aggregate $\mathcal{I}$. According to Theorem 1.4.3 of [78], the arrival curve of the output flow to be fed into the next switch is given by:

$$\begin{aligned} \alpha^*_{\mathcal{I}}(t) &= (\alpha_{\mathcal{I}} \oslash \beta)(t) \\ &= \sup_{u \geq 0}\{\alpha_{\mathcal{I}}(t+u) - \beta(u)\}, \end{aligned} \tag{6.9}$$

where $\oslash$ represents the deconvolution operation under min-plus algebra. Note that in order to avoid data overflow, the service rate should not be less than the sum of arrival rates of VLs in $\mathcal{I}$. Obviously, when $t \geq 0$, (6.9) achieves the maximum value with $u = 0$ and then we have:

$$(\alpha_{\mathcal{I}} \oslash \beta)(t) = \alpha_{\mathcal{I}}(t).$$

Then, the instantaneous burst of aggregate traffics is at most $(\alpha_{\mathcal{I}} \oslash \beta)(0) = \sum_k \sigma_k$, and thus the output flow $\alpha^*_{\mathcal{I}}(t)$ ($t > 0$) is still constrained by the original arrival curve $\alpha_{\mathcal{I}}(t)$.

However, most problems of practical interest cannot be solved as simply as the above case. Some VLs with different destinations need to be subtracted from the aggregate $\mathcal{I}$. Thus, a major difficulty is to derive the arrival curve for the remaining flows of the aggregate.

Suppose that $\mathcal{I}_1$ is a sub-aggregate of VLs subtracted from $\mathcal{I}$, and $\mathcal{I}_2 = \mathcal{I} \setminus \mathcal{I}_1$ represents the sub-aggregate of the remaining VLs. As mentioned before, the staircase arrival curve is tighter than the affine arrival curve. Therefore, if the sub-aggregate $\mathcal{I}_1$ is constrained by

$$\alpha_{\mathcal{I}_1}(t) = \sum_{j \in \mathcal{I}_1} \alpha_{T_j, T_j},$$

it is also constrained by the affine arrival curve

$$\alpha'_{\mathcal{I}_1}(t) = \sum_{j \in \mathcal{I}_1} \alpha_{\rho_j, \sigma_j}.$$

Due to the property of FIFO scheduling, $\mathcal{I}_2$ will be served after $t_0 = \frac{\sum_{j \in \mathcal{I}_1} \sigma_j}{C}$ with a guaranteed service rate $C - \sum_{j \in \mathcal{I}_1} \rho_j$, where C is the service rate offered for the aggregate $\mathcal{I}$ and $\rho_j = \frac{\sigma_j}{T_j}$. Therefore, the service curve for sub-aggregate $\mathcal{I}_2$ can be expressed by

$$\beta_{\mathcal{I}_2}(t) = \left(C - \sum_{j \in \mathcal{I}_1} \rho_j \right) (t - t_0)^+ .$$

When $t \geq 0$, we have:

$$\begin{aligned} \alpha^*_{\mathcal{I}_2}(t) &= (\alpha_{\mathcal{I}_2} \oslash \beta_{\mathcal{I}_2})(t) \\ &= \alpha_{\mathcal{I}_2}(t + t_0). \end{aligned}$$

The details of the calculation are deferred to Appendix. The output flow is constrained by $\alpha_{\mathcal{I}_2}(t + t_0)$, $t \geq 0$, instead of $\alpha_{\mathcal{I}_2}(t)$. In order to have a better understanding, an illustration is given in Figure 6.6.

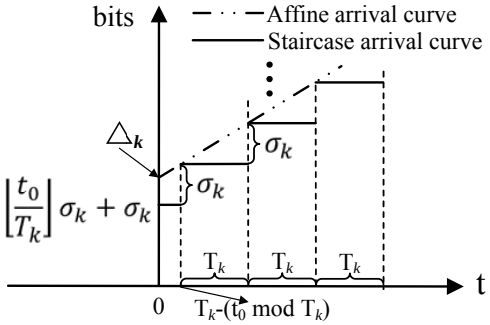


Figure 6.6 The arrival curve for the output of VL_k, $\tau = t_0 + T_k$.

Clearly, the output of VL_k has a staircase arrival curve

$$\begin{aligned}\alpha_{T_k,t_0+T_k}(t) &= \left\lfloor \frac{t+t_0}{T_k} \right\rfloor \sigma_k + \sigma_k \\ &\leq \frac{\sigma_k}{T_k} t + \left(\frac{t_0}{T_k} + 1 \right) \sigma_k \\ &= \rho_k t + \Delta_k,\end{aligned} \tag{6.10}$$

where $t, t_0 \geq 0$, $\rho_k = \frac{\sigma_k}{T_k}$, and $\Delta_k = (\frac{t_0}{T_k} + 1)\sigma_k$. Furthermore, the backlog upper bound for the output of VL_k can be obtained by

$$\begin{aligned}B^k_{\max} &= \sup_{s \geq 0} \left\{ [\alpha_{T_k,t_0+T_k}(s) - \beta_k(s)]^+ \right\} \\ &= \sup_{s \geq 0} \left\{ \left[\left\lfloor \frac{s+t_0}{T_k} \right\rfloor \sigma_k + \sigma_k - C_k \times s \right]^+ \right\} \\ &= \max \left\{ \left\lfloor \frac{t_0}{T_k} \right\rfloor \sigma_k + \sigma_k, \left\lfloor \frac{t_0}{T_k} \right\rfloor \sigma_k + 2\sigma_k - C_k \left(T_k - (t_0 \bmod T_k) \right) \right\},\end{aligned} \tag{6.11}$$

where C_k is the service rate of the output of VL_k. As $C_k \geq \rho_k$, we have

$$B^k_{\max} \leq \Delta_k. \tag{6.12}$$

Then the backlog can be improved by $\Delta_k - B^k_{\max}$ compared to the existing results given by, e.g., [78]. Thus, the backlog for the output of sub-aggregation $\mathcal{I}_2$ is bounded by $\sum_{k \in \mathcal{I}_2} B^k_{\max} \leq \sum_{k \in \mathcal{I}_2} \Delta_k$.

6.4.3 End-to-End Delay Analysis

In the following, the staircase arrival curve is employed to obtain a jitter upper bound. Consider a VL aggregate $\mathcal{I}$ served by a constant rate C. Note that in order to avoid data overflow, the service rate must be not less than the sum of arrival rates of VLs, and then we have:

$$\beta(t) = Ct \geq \sum_{k \in \mathcal{I}} (\rho_k t) \geq \sum_{k \in \mathcal{I}} \left\lfloor \frac{t}{T_k} \right\rfloor \sigma_k,$$

where $\rho_k = \frac{\sigma_k}{T_k}$. As the backlog corresponds to the maximum vertical deviation between the arrival curve and the service curve, at any time, the backlog $B(t)$ can be upper bounded by:

$$\begin{aligned} B(t) &\leq \sup_{s\geq 0}\{\alpha_{\mathcal{I}}(s) - \beta(s)\} \\ &= \sup_{s\geq 0}\left\{\sum_k \left\lfloor \frac{s}{T_k} \right\rfloor \sigma_k + \sum_k \sigma_k - Cs\right\} \\ &= \sum_k \sigma_k. \end{aligned} \tag{6.13}$$

In other words, when a frame arrives, there are at most $\sum\limits_k \sigma_k$ bytes in the output buffer waiting for transmission. Therefore, for any VL_j, its worst-case jitter J_Q^j is bounded by:

$$\begin{aligned} J_Q^j &= \sup_{t\geq 0}\left(\inf_{u\geq 0}\{\alpha_{\mathcal{I}}(t) \leq \beta(t+u)\}\right) \\ &= \frac{\sum\limits_{k\in\mathcal{I}} \sigma_k}{C}. \end{aligned} \tag{6.14}$$

The same result as in (6.14) can be found in [80, 78], where the arrival curve assigned to VL_j uses the affine model. The difference is that in the affine model, the assumption that the fresh bit arrives immediately after its burst is made, which leads to a situation where the burst introduces a jitter to subsequent data. By contrast, in the staircase model, the subsequent frame arrives after one BAG in the worst case, which means that the burst does not necessarily introduce additional jitters. Thus, when the condition, $\sum_{k\in\mathcal{I}} \sigma_k/C \leq T_j$, is satisfied, the worst case given in (6.14) can be further improved and bounded by:

$$J_Q^j = \frac{\sum\limits_{k\in\mathcal{I}} \sigma_k}{C} - \frac{\sigma_j}{C} = \frac{\sum\limits_{k\in\mathcal{I}, k\neq j} \sigma_k}{C}. \tag{6.15}$$

In this case, the next frame will not arrive before the current frame is transmitted, and thus the VL_j's burst does not introduce a jitter. Therefore, the jitter can be improved by $\frac{\sigma_j}{C}$ compared to the existing results given by, e.g., [78].

Furthermore, the end-to-end service curve for one VL passing cascaded switches can be concatenated by using the convolution under the min-plus algebra defined as follows:

$$(f \otimes g)(t) = \inf_{0\leq s\leq t}\{f(t-s) + g(s)\}. \tag{6.16}$$

By using the equivalent model, a tighter bound is obtained, which is known as "Pay Bursts Only Once" [78]. In the following, a simple example is given to show the calculation in detail.

As shown in Figure 6.7, two VL aggregates, $\mathcal{I}$ and $\mathcal{K}$, are delivered from ES1 and ES2,

respectively. Suppose that the service curve offered by each scheduler is $\beta = Ct$. Suppose further that each VL is constrained by $\alpha_{T,\sigma}(t)$, in which T represents its BAG and its burst is constrained by σ. After reaching Switch 1, $\mathcal{I}$ is separated into two sub-aggregates, $\mathcal{I}_1$ and $\mathcal{I}_2$, corresponding the flows to two different destinations. Consider an end-to-end jitter of a flow of VL_k, $k \in \mathcal{K}$.

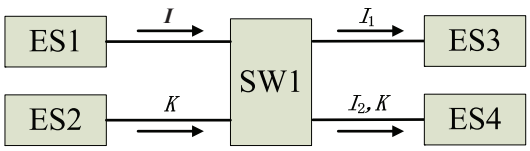


Figure 6.7 An example of end-to-end jitter analysis for one VL of interest.

Due to the FIFO scheduling, the service curve offered by ES2 for VL_k can be expressed as $\beta_k^{(1)} = C_k^{(1)}(t - \tau_k^{(1)})^+$, where $C_k^{(1)} = C - \sum_{j\in\mathcal{K}, j\neq k} \frac{\sigma_j}{T_j}$ and $\tau_k^{(1)} = \frac{\sum_{j\in\mathcal{K}, j\neq k} \sigma_j}{C}$. Based on the analysis above in Section 6.4.2, the service curve offered by Switch 1 for VL_k is $\beta_k^{(2)} = C_k^{(2)}(t - \tau_k^{(2)})^+$, where $C_k^{(2)} = C - \sum_{j\in\mathcal{K}, j\neq k} \frac{\sigma_j}{T_j} - \sum_{i\in\mathcal{I}_2} \frac{\sigma_i}{T_i}$ and $\tau_k^{(2)} = \frac{\sum_{j\in\mathcal{K}, j\neq k} \sigma_j}{C} + \frac{\sum_{i\in\mathcal{I}_2} B^i_{\max}}{C}$. Obviously, the bursts of VLs in $\mathcal{K}$, excluding σ_k, are taken into account twice in both ES2 and Switch 1. As the service rate offered by these two schedulers is identical, we can further have $\beta_k'^{(2)} = C_k^{(2)}(t - \tau_k'^{(2)})^+$, where $\tau_k'^{(2)} = \frac{\sum_{i\in\mathcal{I}_2} B^i_{\max}}{C} \leq \frac{\sum_{i\in\mathcal{I}_2} \Delta_i}{C}$. By applying the min-plus convolution, the end-to-end service curve is given as:

$$\begin{aligned}
\beta_k(t) &= \left(\beta_k^{(1)} \otimes \beta_k'^{(2)}\right)(t) \\
&= \min\left\{C_k^{(1)}, C_k^{(2)}\right\}\left(t - \tau_k^{(1)} - \tau_k'^{(2)}\right)^+ \\
&= C_k^{(2)}\left(t - \tau_k^{(1)} - \tau_k'^{(2)}\right)^+.
\end{aligned}$$

Therefore, the end-to-end jitter for VL_k is

$$J_{e2e}^k = \frac{\sigma_k}{C_k^{(2)}} + \tau_k^{(1)} + \tau_k'^{(2)}.$$

Based on the analysis in Section 6.4.3 with staircase arrival curve, if $J_{e2e}^k \leq T_k$, the end-to-end jitter can be improved by $\frac{\sigma_k}{C_k^{(2)}}$. Thus, a tighter result is:

$$J_{e2e}'^k = \tau_k^{(1)} + \tau_k'^{(2)}.$$

By incorporating the convolution operation for the service curve under concatenation, a tighter end-to-end jitter upper bound for the VL of interest is obtained.

6.5 Approaches to Eliminate the Occurrence of Frame Sequence Inversion

As shown in (6.6), to avoid the SN inversion, the sum of jitter upper bound and transmission latency difference has to be constrained within one BAG. In addition to the use of the tighter jitter upper bound estimation presented in the previous section, this section addresses the possible solutions for further reducing the jitter and the transmission latency difference.

6.5.1 Local Synchronization

The jitter upper bound estimation is based on the worst case, where it is assumed that the frames in all the VLs arrive simultaneously. However, this situation will not happen when some applications are executed sequentially on a single processor, which is common in practice. For example, the AFDX ESs are often paired with the ARINC 653 operating system (OS). Thus the frames of certain VLs are produced with a static and strictly periodic manner on distinct pre-defined time slots. Therefore, LS is a possible solution to mitigate the jitter by exploring the periodic VL characteristics in the source ES. Relevant research on LS in ESs can be found in [77] and [88]. It is proposed in [77] to reduce the end-to-end delay by taking into account partition scheduling, which helps to eliminate impossible scenarios by introducing a correlation between the release of VLs in each ES. In [88], all VLs are assumed to be periodic and the offsets are assigned to VLs to reduce the end-to-end delay upper bounds. In this section, we further develop this idea while leveraging the staircase arrival curve to improve the results based on [77] and [88].

A strictly periodic VL, e.g. VL_i, can be characterized by a triplet $\{T_i, \sigma_i, O_i\}$, where T_i is the period of the flow and $T_i = \text{BAG}_i$, σ_i is equal to the MFS plus 20 bytes overhead during transmission on physical links, and O_i represents a time offset of the first frame. In the following analysis, the staircase model is applied to assume that an entire frame is released at the start time. In such a model, the jitter of a frame is caused by the residual bytes left for transmission when the frame arrives at the scheduler.

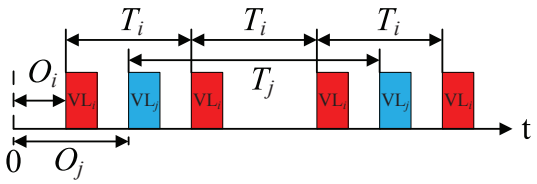


Figure 6.8 An example of two strict periodic VLs with offsets.

First, we just consider the case with two strict periodic VLs as shown in Figure 6.8, in which VL_i starts earlier than VL_j. Note that it can happen that there is more than one frame from

VL_i between two consecutive frames of VL_j. Since T_i is enough for a frame of VL_i to be transmitted, only the adjacent frame ahead of VL_j is taken into account. Furthermore, since the periods are powers of 2, the Greatest Common Divisor (GCD) of the periods corresponds to the operation "min" [49]. Therefore, the release time difference between adjacent frames of the two strictly periodic VLs is given by

$$D_{\text{diff}} = \begin{cases} |O_i - O_j| \bmod \min\{T_i, T_j\}, \ or \\ \min\{T_i, T_j\} - (|O_i - O_j| \bmod \min\{T_i, T_j\}). \end{cases} \tag{6.17}$$

If the condition $\frac{\max\{\sigma_i, \sigma_j\}}{C} \leq D_{\text{diff}}$ is satisfied, the two VLs have no influence on each other, although they share the output port of the same source ES. The reason is that there is enough time for the current frame, either in VL_i or in VL_j, to be delivered before the generation of a new frame from the other VL. Note that at the scheduler of the source ES, it is possible that the frames belonging to VL_i and VL_j are delayed mutually. However, it is due to the delay propagation introduced by other VLs, but not by VL_i or VL_j. The situation is the same in the switches along the transmission path. Once the VLs are delivered from the source ES, they are serialized. If the frame dispatched earlier does not experience any congestion with other VLs in all switches along its path, it will never interfere with a frame released later. Although jitter may be introduced by a frame dispatched earlier, when congestion happens, it is due to the jitter propagation caused by other VLs, similar to the case in ES. Obviously, LS contributes to reduce the jitter, as the number of interfering VLs to take into account is diminished.

In addition, if the order of the two VLs is fixed, e.g. VL_i always starts ahead of VL_j, then the requirements can be relaxed. In this scenario, VL_j has no influence on VL_i if the condition $\frac{\sigma_j}{C} \leq (\min\{T_i, T_j\} - ((O_j - O_i) \bmod \min\{T_i, T_j\}))$ holds and VL_i has no influence on VL_j if the condition $\frac{\sigma_i}{C} \leq ((O_j - O_i) \bmod \min\{T_i, T_j\})$ can be met. This method can be extended when a set of strictly periodic VLs (>2) is considered and the corresponding analysis is given in the following.

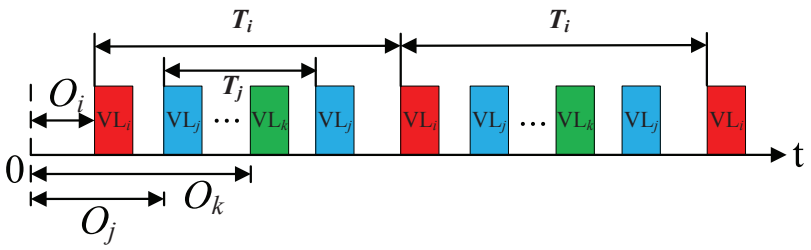


Figure 6.9 An example of multiple strictly periodic VLs with offsets.

For instance, as shown in Figure 6.9, VL_i is the VL of interest and it always arrives ahead of the other VLs. In a period of T_i, there may be more than one frame belonging to the other VLs, e.g. VL_j. Denote by $D_{i,j}$ the release time difference between adjacent frames of VL_i and VL_j, where the frame of VL_j is ahead of that of VL_i. According to (6.17), $D_{i,j} \leq \min\{T_i, T_j\}$. Then the number of VL_j within T_i is upper bounded by

$$N_j = \left\lceil \frac{T_i - D_{i,j}}{T_j} \right\rceil. \tag{6.18}$$

For each pair (VL_i,$\mathrm{VL}_j(q)$), $q = 1, \ldots, N_j$, the release time difference is computed individually and stored in descending order. Likewise, a set composed by all pairs of VLs can be obtained. After reordering the release time difference, this set can be given in the form of $\{\ldots, D_{i,k}^{(l-1)}(1), D_{i,j}^{(l)}(1), D_{i,i}^{(l+1)}(1)\}$, where $l = \sum_{j \neq i} N_j$ is the total number of frames between two consecutive frames of VL_i. In order to simplify the notation, we only use the superscript notation. Therefore, $\sigma^{(l)}$ is associated with VL_j, $\sigma^{(l-1)}$ is associated with VL_k, and so on.

Let $M_i^{(l)}$ be the residual bytes left for transmission in the worst case when VL_i arrives. In other words, $M_i^{(l)}$ is the total number of bytes that contributes to the jitter of VL_i in the worst-case scenario. Then we have

$$M_i^{(l)} = \left[M_i^{(l-1)} + \sigma^{(l)} - \left(D^{(l)} - D^{(l+1)} \right) C \right]^+, \tag{6.19}$$

where $M_i^{(0)} = 0$. If $M_i^{(l)}$ can be reduced by applying LS, the introduced jitter is mitigated accordingly.

To illustrate the effect of LS, we consider an example of 3 VLs with σ=1500 bytes and a BAG of 1 ms. Their offsets are O_1=0, O_2=100 μs and O_3=200 μs, respectively. Then the set of release time difference is given in Table 6.1. Based on (6.18), it can be obtained that $l = 2$. By considering LS, the residual bytes when VL1 arrives is given by:

$$\begin{aligned} M_1^{(2)} &= \left[M_1^{(1)} + \sigma^{(2)} - (D^{(2)} - D^{(3)})C \right]^+ \\ &= \left[\left[M_1^{(0)} + \sigma^{(1)} - (D^{(1)} - D^{(2)})C \right]^+ + \sigma^{(2)} - (D^{(2)} - D^{(3)})C \right]^+ \\ &= 0, \end{aligned}$$

where $\sigma^{(2)} = \sigma^{(1)} = 1500$ Bytes, $D^{(1)} = 900\mu$s, $D^{(2)} = 800\mu$s, $D^{(3)} = 0$, and $C = 12.5$ MBytes/s. Similarly, we can get $M_2^{(2)} = 250$ Bytes and $M_3^{(2)} = 500$ Bytes for VL2 and VL3, respectively. The residual data for VL2 and VL3 can be further mitigated by properly adjusting the offsets. In contrast, by applying the conventional approaches without

LS, the residual bytes for each VL are $M_1^{(2)} = M_2^{(2)} = M_3^{(2)} = 3\sigma = 4500$ Bytes in the worst case. In this example, the residual bytes are significantly reduced with LS.

Table 6.1 Time Interval between Frames

VL Pairs (i, j)	1, 2	1, 3	2, 1	2, 3	3, 1	3, 2
$D_{i,j}$ (μs)	900	800	100	900	200	100

It is shown in the above analysis that LS contributes to reduce the residual bytes for a periodic VL. Consequently, the jitter for the periodic VL is mitigated so that the incidence of frame sequence inversion never happens in that case. The other periodic VLs also benefit from this approach. In fact, the jitters for the other periodic VLs may be further mitigated by properly allocating the offsets of periodic VLs. Moreover, the aperiodic VLs can also benefit from the LS as the frames in different periodic VLs cannot arrive simultaneously. Thus, the number of interfering VLs or the amount of interfering backlog can be reduced. Compared with the approach in [88], our jitter upper bounds are obtained by analyzing the residual number of bytes with respect to LS, instead of using the safe arrival curve of the aggregated flows. Therefore, tighter upper bounds can be achieved. For example, the end-to-end delay upper bound of $v1$ from $e1$ to $e6$ in the case study presented in [88] can be reduced to 96μs from 116μs due to the fact that the VLs $v1$ and $v2$ have no influence on each other according to our model. It is worth noting that LS can also help to eliminate certain impossible scenarios in switches to further improve jitter estimation as presented in [88]. This feature is taken into account in the case study presented in Section 6.6.

6.5.2 Transmission Latency Difference Minimization

It can be seen from (6.6) that the transmission latency difference between two continuous frames defined in (6.5) is another factor that may cause sequence inversion. Thus, we consider a scheme aiming at reducing the second term on the left-hand side of the inequality (6.6) by TLDM.

In traditional delay analysis, much attention has been paid to the MFS, as the minimum length has no effect on the jitter upper bounds. Normally, the default minimum length predetermined by the specification is assigned to each VL. In fact, this makes the transmission latency difference even larger according to (6.5). In the worst case, the frames with the MFS and the frames with minimum length are delivered alternately as shown in Figure 6.10. In this scenario, half the received frames experience the worst-case transmission latency, which increases the occurrence probability of the sequence inversion phenomenon.

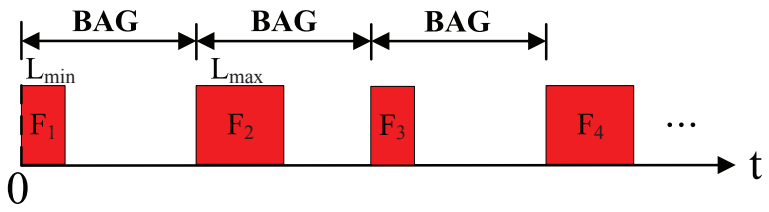


Figure 6.10 An example of transmission latency difference in the worst case.

Based on (6.5), for a predefined VL routing scheme, the transmission latency mitigation can be formulated as an optimization problem aimed at minimizing the maximum size difference between two continuous frames:

$$\min_{i\in[0,\,255]} \max\left[L(i) - L(i+1)\right]^{+}, \tag{6.20}$$

where the wrap-around operation $i+1$ is defined in (6.1). It can be further simplified as the following problem:

$$\min_{i\in[0,\,255]} \left(L_{\max}(i) - L_{\min}(i+1)\right). \tag{6.21}$$

Obviously, the optimal value of (6.20) and (6.21) is zero. It can be achieved when every frame in a VL is set to the identical frame size, $L_{\max} = L_{\min}$. However, the configuration for each VL in practice cannot be simply assigned in such a way due to diverse requirements and data source types. In this case, the transmission latency difference can be mitigated by properly selecting the value of $L_{\min}$, and thus both (6.20) and (6.21) are upper bounded by $L_{\max} - L_{\min}$. Even though the optimum of (6.20) or (6.21) is not achieved, the TLDM helps control the transmission latency difference by carefully selecting $L_{\min}$. Therefore, this approach contributes to satisfy the inequality (6.6) so that the sequence inversion can be avoided.

To illustrate how TLDM contributes to reduce the transmission latency difference between two consecutive frames, we consider a case in which a VL has a MFS of 600 bytes and a default minimum length of 64 bytes. The VL traverses 2 switches to reach its destination. Then the transmission latency difference can be up to $\dfrac{(600-64)\times 8}{C} \times 3 = 128.64\ \mu s$, if $C = 12.5$ MBytes/s. When $L_{\min}$ is 500 bytes, the upper bound of transmission latency difference can be reduced to 24 μs, less than 20% compared with 128.64 μs. The optimal value of transmission latency difference is zero and it can be achieved with $L_{\min} = 600$ bytes. Since the minimum frame length is not used during the worst case delay analysis, enforcing $L_{\max} = L_{\min}$ does not change the performance of the network in the worst case. This example confirms that the specification of frame size has an impact on the transmission reliability and

should be carefully designed.

Design rules allowing improving transmission reliability can be generally given as follows:

— assign identical or similar frame size for all the frames in a VL;

— if the message is too large and needs to be fragmented, assign an equal size to each fragment;

— if Sub-VL aggregation is performed as in [83] to optimize bandwidth utilization, the pre-processing is required first to assort Sub-VLs with similar frame size into a group. Then Sub-VL aggregation strategy is applied to each group to avoid large transmission latency differences.

6.6 Case Study

In this section, the proposed approaches are illustrated by a case study with a network shown in Figure 6.11, which is adapted from a benchmark configuration reported in [20, 88, 8, 62, 71] while including more VLs. The VL parameters are specified in Table 6.2 and Table 6.3, in which VL1-8 are strictly periodic VLs and each period T is equal to its BAG.

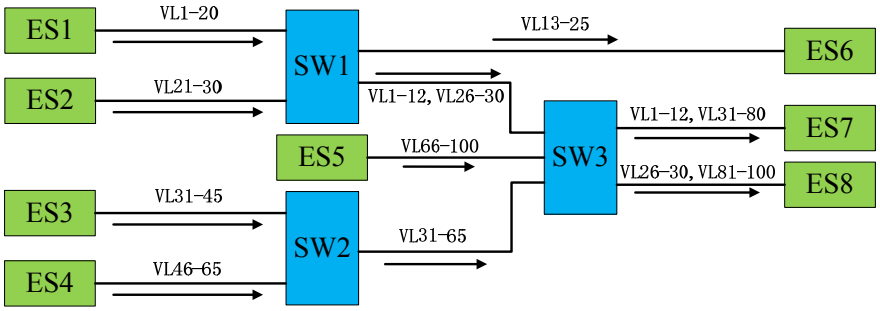


Figure 6.11 An example of VL management in source ESs and the end-to-end transmission schematic.

Table 6.2 Parameters of strictly periodic VLs

VL	1	2	3	4	5	6	7	8
BAG (ms)	1	4	4	2	4	2	4	64
σ (byte)	620	84	520	820	320	140	1020	520
O (ms)	0.1	0.5	0.5	0.5	0.8	0.8	1.5	1.5
Number of Hops	3	3	3	3	3	3	3	3

Table 6.3 Parameters of aperiodic VLs

VL	9-12	13-20	21-25	26-30	31-35	36-40
BAG (ms)	2	1	4	16	2	1
σ (byte)	250	84	620	480	100	84
Number of Hops	3	2	2	3	3	3
VL	41-45	46-55	56-65	66-80	81-90	91-100
BAG (ms)	2	2	4	2	1	8
σ (byte)	260	180	200	84	100	320
Number of Hops	3	3	3	2	2	2

In this case study, we assume that the physical link offers a constant rate $C = 12.5$ MBytes/s. Suppose that VL1 is the data flow of interest. First, the end-to-end jitter upper bound obtained from the affine model is 1.283 ms. Then the staircase model presented in Section 6.4 is applied. As the LS is not applied at this step, VL1 is assumed to be influenced by other VLs that share the same output ports either in source ES or in switches. The obtained result is reduced to 1.263 ms, benefiting from the improvement of the backlog upper bound estimation as presented in (6.13). In addition, the minimum frame size of VL1, $L_{\min}$, is assigned to 64 Bytes as default. Since VL1 traverses two switches in its communication path, the transmission latency difference in the worst case can be obtained with $(L_{\max} - L_{\min})/C \times n$, where $n = 3$. In this scenario, the maximum transmission latency difference is 128.64μs, which is more that 10% of its BAG. Considering a transmission latency difference of 128.64μs, the worst-case delay difference can be up to 1.392 ms, which clearly exceeds its BAG, the safe upper bound. In the following, the approaches presented in Section 6.5 are applied step by step to mitigate the delay differences.

The LS focuses on the strictly periodic VL1-8. According to Table 6.2, VL1 is always ahead of VL2-8, then the temporal interval between each pair is calculated based on (6.17) and listed in Table 6.4, in which the required transmission time for VL2-8 is also given. We further compute the residual bytes, which may introduce a jitter into VL1. The calculation is based on (6.19). In this example, $M_1^{(l)}$=0, where l=7. In other words, VL2-8 have sufficient time to be delivered before the arrival of VL1 and hence, they have no impact on VL1 in terms of jitter. The jitter upper bound can be further improved by reducing the number of involved VLs. The obtained result is 0.989ms, which is less than its BAG. In this case, its burst does not introduce jitter in the worst case when the staircase arrival curve model is employed. Therefore, the end-to-end jitter could be reduced by 0.081ms, and then the upper bound becomes 0.908ms.

Table 6.4 Temporal Interval between Frames and the Transmission Time Requirement (in μs)

VL Pairs (i, j)	1, 2	1, 3	1, 4	1, 5	1, 6	1, 7	1, 8
$T_{\min}-(O_j-O_i)$	600	600	600	300	300	600	600
σ_j/C	6.72	41.6	65.6	25.6	11.2	81.6	41.6

Till now, although a large improvement has been achieved, the requirement cannot be met when considering the fixed transmission latency difference of 128.64μs in the worst case. The sum of the jitter and the latency difference is 1.037ms, which is very close to the safe upper bound.

Thereafter, the TLDM is applied. As illustrated in Section 6.5.2, the fixed transmission latency difference can be improved by more than 80% if $L_{\min}$ is 500 bytes, and then the transmission delay difference is 0.933ms<1ms. The optimal result for transmission latency difference is zero, when the VL guarantees that all the frames have an identical frame size. With either of the two improved configurations, it can be verified that the transmission delay difference will not exceed the BAG and the sequence inversion will never happen for VL1.

Finally, the delay differences in the worst case for all other VLs are computed using the classical affine model, the staircase model, and the approach based on LS and TLDM, respectively. As shown in Figure 6.12, there is a potential risk of failures for the redundant transmission of VL1 and VL36-VL40, as the delay differences obtained based on the affine model are larger than their BAGs (1ms). The application of the staircase model results in more accurate estimates, which are small enough to ensure that sequence inversion in VL36-VL40 will not occur. However, VL1 is still unsafe. When the approaches of LS and TLDM are further applied, the delay differences for all the VLs meet the requirements, and the sequence inversion phenomenon has been avoided. The improvement with TLDM is achieved under the condition that the frame size difference is restricted within 100 Bytes.

It is worth noting that ultimately, one can assign a VL to each application. Therefore, the constraints on frame size difference can always be satisfied by adding VLs. In essence, this amounts to a tradeoff between the reliability and the bandwidth utilization efficiency.

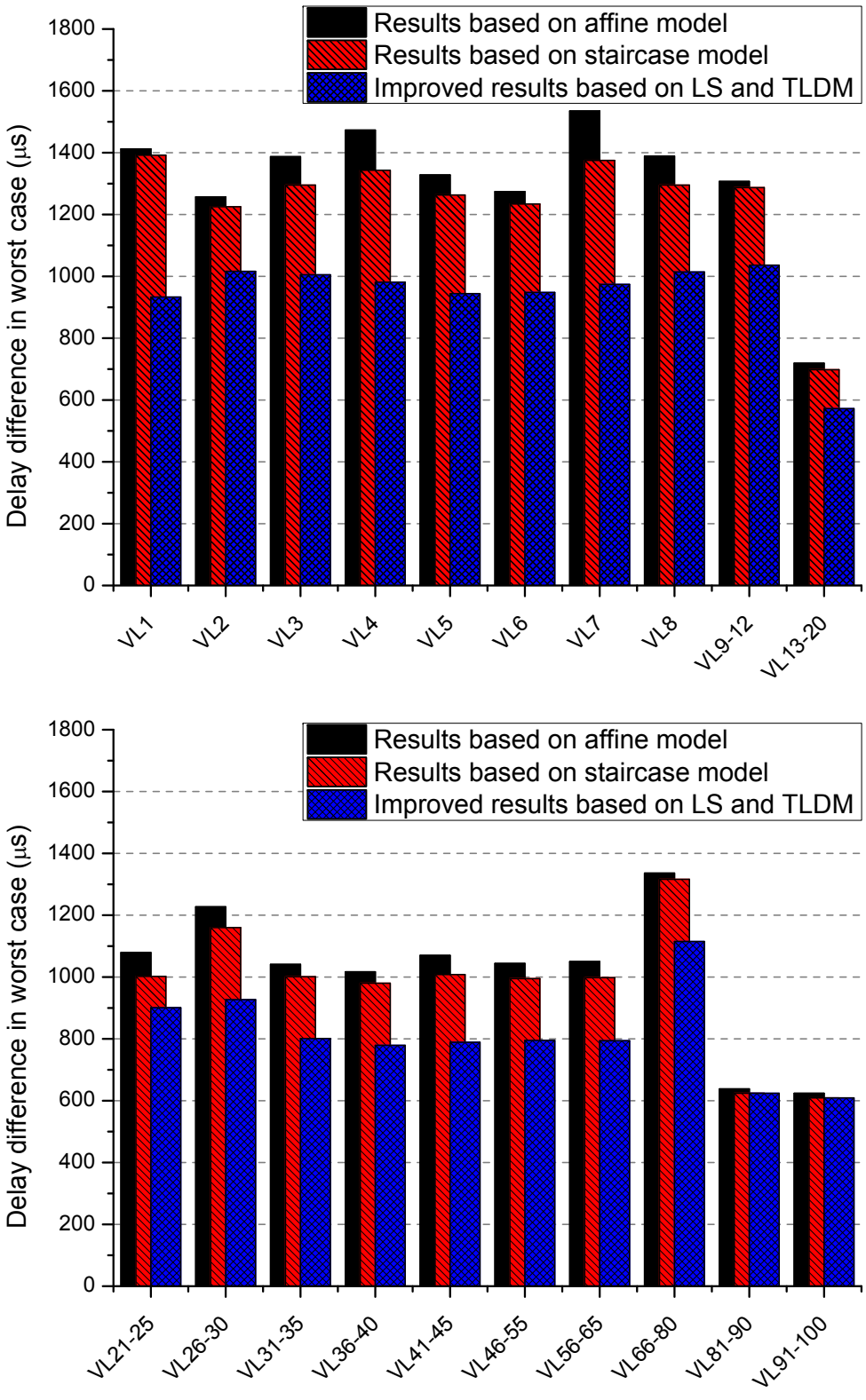


Figure 6.12 Delay differences in the worst case for all the VLs.

6.7 Conclusion

In this paper, a potential failure in the redundant transmission management of AFDX networks is addressed, and a quantitative analysis of this phenomenon is carried out. It has been found that the main reason is due to the sequence inversion phenomenon in redundant networks caused by the jitter and frame size difference. Then a more accurate jitter estimation is first proposed based on the staircase arrival curve. In order to eliminate the potential failure, two approaches are developed, which allow tightening jitter estimation by reducing the number of VLs involved and diminishing transmission latency differences. A case study is carried out to illustrate the proposed approaches. The results confirm that the developed approaches are feasible and effective.

6.8 Appendix

As stated in Section 6.4.2, the service curve for sub-aggregate $\mathcal{I}_2$ is given by

$$\beta_{\mathcal{I}_2}(t) = \left(C - \sum_{j\in\mathcal{I}_1}\rho_j\right)(t-t_0)^+,$$

where $t_0 = \dfrac{\sum_{j\in\mathcal{I}_1}\sigma_j}{C}$. Then we have:

$$\begin{aligned}\alpha^*_{\mathcal{I}_2}(t) &= (\alpha_{\mathcal{I}_2} \oslash \beta_{\mathcal{I}_2})(t)\\ &= \sup_{u\geq 0}\{\alpha_{\mathcal{I}_2}(t+u) - \beta_{\mathcal{I}_2}(u)\}\\ &= \max\left\{\sup_{0\leq u\leq t_0}\{\alpha_{\mathcal{I}_2}(t+u)\}, \sup_{u>t_0}\{\alpha_{\mathcal{I}_2}(t+u) - \beta_{\mathcal{I}_2}(u)\}\right\}.\end{aligned}$$

According to the assumption,

$$\begin{aligned}\left(C - \sum_{j\in\mathcal{I}_1}\rho_j\right)t &\geq \sum_{k\in\mathcal{I}_1}\rho_k t,\\ &\geq \sum_{k\in\mathcal{I}_1}\left\lfloor\frac{t}{T_k}\right\rfloor\sigma_k.\end{aligned}$$

Obviously, when $t \geq 0, \alpha^*_{\mathcal{I}_2}(t) = \alpha_{\mathcal{I}_2}(t+t_0)$.

CHAPTER 7 GENERAL DISCUSSION

The specification of ARINC664 part 7 has brought a number of improvements, such as the introduction of the concept of VL and the employment of redundancy mechanisms, to overcome the shortcomings of the IEEE 802.3 Ethernet protocol. Nevertheless, there are still some potential problems in AFDX as pointed out in this thesis. Therefore, the main focus of the thesis is to enhance AFDX to achieve a fully deterministic and reliable network.

The initial investigation relates to non-determinism with respect to frame arrival, whose uncertainty introduces a problem in terms of real-time fault detection. In order to improve the determinism of AFDX networks, a mechanism based on frame insertion is proposed. As this enhancement is achieved at the expense of network load increase, a Sub-VL aggregation strategy is further developed in order to mitigate the overhead due to frame insertion. This strategy is formulated as a multi-objective optimization problem considering the trade-off between load increase and the delay introduced by Sub-VL aggregation. Then three algorithms, namely a brute force algorithm, a greedy algorithm, and a greedy algorithm with pre-processing, are proposed and investigated to solve the Sub-VL aggregation optimization problem. The brute force algorithm can reach the global optimal solution, while it is suitable only for small numbers of considered Sub-VLs. The greedy algorithm and its variation with pre-processing are suitable for large size problems, although they may lead to local optimums. Simulations are carried out to illustrate the feasibility of the proposed frame insertion method and to validate the performance of developed algorithms. The results show that the load increase can be dramatically reduced and the delay introduced by Sub-VL aggregation can be mitigated with a relaxed δ-constraint. Finally, the non-determinism can be removed by the introduced mechanism, which meanwhile enables real-time fault detection in destination ESs. It is worth noting that the focus of this work is put on the configuration in source ESs. The impact of Sub-VL aggregation with frame insertion has to be carefully evaluated against the overall performance requirements for specific applications in the design of AFDX networks. As this work mainly focuses on determinism enhancement, a formal analysis of the algorithm complexity is reserved as the future work to explore better results for Sub-VL aggregation.

Following the investigation into sources of non-determinism of AFDX networks, the concept of incorporating the performance analysis into a quantitative reliability assessment is proposed. This leads to introducing end-to-end delay violations as a form of failure, which may allow for the adoption of probabilistic upper bounds in AFDX network certification. The

Stochastic Network Calculus is applied to compute upper bounds with various probability limits. In contrast to the deterministic analysis, stochastic approaches may offer more realistic and tighter bounds by capturing the probabilistic nature of networks. Furthermore, the reliability analysis of AFDX networks is performed with an investigated example by using the FTA technique, which shows how to integrate VL delay violation probabilities into the reliability analysis. A case study is carried out to demonstrate the safety of the probabilistic bounds. It is observed that some system reliability requirement can still be met, even considering a certain probability of VL delay violations. Moreover, the results show that the probabilistic upper bounds are significantly less pessimistic than the deterministic ones, which can facilitate network design by offering a larger margin regarding delay requirements for delay-sensitive applications. In fact, tighter upper bounds for VLs also contribute to reducing the occurrence probability of redundant transmission failures. It is worth noting that for simplicity the scheduling policy considered in this work is First-Come, First-Served. Different scheduling strategies may significantly impact jitter probability distributions.

In this thesis, a potential failure in the redundant transmission management of AFDX networks is also addressed, and a quantitative analysis of this phenomenon is further carried out. It has been found that the main reason is due to the sequence inversion phenomenon in redundant networks caused by the jitter and frame size difference. Then a more accurate jitter estimation is first proposed based on the staircase arrival curve, which is tighter than the affine arrival curve. Furthermore, two approaches are developed, which allow tightening jitter estimation by reducing the number of VLs involved and diminishing transmission latency differences. One of the two approaches is based on LS to tighten jitter estimation by reducing the number of VLs involved. The other exploits the notion of TLDM to diminish transmission latency differences. It is shown that these two methods can be applied separately or in combination, and that combining them gives the best results. Finally, a case study is carried out to illustrate the proposed approaches. The results confirm that the developed approaches are feasible and effective. Currently, the comparison of the analysis approaches is mainly based on their accuracies. In future work, the analysis complexity will be investigated, based on which suitable tools will be developed for practical applications.

CHAPTER 8 CONCLUSION AND PROSPECTIVE

8.1 Conclusion

This thesis proposed several approaches related to determinism enhancement, reliability assessment and reliability improvement. These characteristics are treated as quantitative performances that can be obtained by carrying out suitable analysis. The methods proposed and explored in this thesis are validated through case studies and the reported results confirm their feasibility and applicability.

Chapter 1 explained the context of the present research project, introduced the performance evaluation and reliability assessment of avionics networks, and reviewed existing techniques for timing analysis in AFDX networks. Then the research contributions were highlighted.

Chapter 2 discussed the evolution of avionic networks and some of the most employed avionic communication network protocols, i.e., ARINC 429, MIL-STD-1553B, ARINC 825, TTEthernet, and AFDX. A comparison was made to show the outstanding performance of AFDX.

Chapter 3 explains the details of the analysis tools, namely deterministic Network Calculus and Stochastic Network Calculus, which have been applied in AFDX network performance analysis. Then, the multi-objective optimization problem is introduced and approaches for achieving optimal solutions are presented. Finally, Fault Tree Analysis that is one of the most prominent analysis techniques for quantitative reliability assessment, is reviewed. All these tools are employed in this thesis to improve the performance of AFDX networks.

Chapter 4 presented the article entitled "Determinism Enhancement of AFDX Networks via Frame Insertion and Sub-Virtual Link Aggregation". In this chapter, a mechanism was proposed to enhance determinism of AFDX networks via frame insertion. Meanwhile, in order to mitigate network load increase due to frame insertion, a Sub-Virtual Link aggregation strategy was introduced. Finally, the problem was formulated as a multi-objective optimization problem considering the trade-off between traffic load and delay due to Sub-VL aggregation. Three algorithms have been developed to find solutions to the optimization problem. Experiments were carried out to verify the proposed mechanism. A real-time system simulation software, TrueTime, was utilized to validate the proposed mechanism considering Sub-VL aggregation.

Chapter 5 is based on the article "Incorporating Performance Analysis into Reliability Assessment for Avionics Full-Duplex Switched Ethernet Networks". In this chapter, an approach was introduced to incorporate performance analysis into reliability assessment by considering

the delay violation as a type of failure and establishing a corresponding reliability assessment model. Then, the well-known FTA technique was employed to perform reliability assessment while taking into account the failures due to delay violations. SNC was also applied to compute the upper bounds with various probability limits. This approach is illustrated with a case study, and the results confirmed that the overall system reliability requirement can be met with less pessimistic probabilistic performance constraints. Furthermore, a means of specifying the performance requirements based on tighter bounds associated with probability budgets was provided in order to explore the fault tolerance capabilities of redundant mechanisms.

Chapter 6 introduces the article titled "Reliability Enhancement of Redundancy Management in AFDX Networks". This chapter focused on the phenomenon of sequence inversion, which may induce failures in spite of redundant transmission in AFDX networks. Such failures degrade the network reliability. A mathematical analysis was provided with conditions on the occurrence of this phenomenon. The main sources leading to sequence inversion are due to the jitter and the transmission latency difference between two successive frames. Several solutions that allow avoiding the occurrence of sequence inversion have been developed. The proposed approaches are illustrated through an AFDX network example. This work is a contribution to the efforts aimed at enhancing the determinism and the reliability of AFDX networks to render this promising technology applicable for mission-critical avionics systems. The three papers listed above are claimed as contributions of this thesis.

8.2 Future Work Directions

8.2.1 Analysis of Scheduling Policy

Scheduling policies are employed to control traffic flows and guarantee the Quality of Service (QoS). Since there exist multiple scheduling policies, an analysis is required to make sure which policy offers the best performance. For simplicity, the FIFO (or FCFS) scheduling policy is applied as the default configuration in this thesis to perform various analysis. Thus, it is possible to obtain tighter bounds by further integrating other scheduling policies such as non-preemptive fixed priority policy. For example, in the study of Chapter 5 and Chapter 6 fixed priority policy can be applied, and high priority can be assigned to the VLs associated with delay-sensitive applications. According to the principle of fixed priority policy, the frames with the highest priority, which are currently ready for transmission, will be delivered earlier than the ones with lower priorities. In this case, the end-to-end delay of VLs with high priority can be reduced. Consequently, the performance and the reliability for the

corresponding VLs can be improved.

8.2.2 Jitter Analysis in Switches

In AFDX networks, switches have an essential role influencing the overall network performance. In Chapter 6, an approach based on LS has been applied in ESs, which results a tighter jitter estimation. In fact, this approach may also have an impact on switches considering the effect of frame serialization after aggregation. Thus, further investigation can be made to perform a more accurate jitter estimation in switches. Besides, the introduction of scheduling policy, e.g., fixed priority scheduling, will influence the jitter estimation in switches too. As a result, future work could also focus on the integration of scheduling policies in switches to obtain tighter jitter upper bounds.

8.2.3 Performance Analysis Under a Mixed Network Architecture

The current communication system of new generation aircrafts, e.g., A350, is typically composed of a mixed network architectures based on an AFDX backbone and other ancillary networks. Thus, the end-to-end delay experienced by a frame is the sum of delays encountered in all crossed networks and components. Therefore, trade-off can be made to balance the delay in each portion of the network. In this context, characterizing and optimizing performances under a mixed network architecture is regarded as another possible future research direction.

BIBLIOGRAPHY

[1] The A380 by-the-numbers: Impressive on all counts. http://www.airbus.com/news-events/news-events-single/detail/the-a380-by-the-numbers-impressive-on-all-counts.

[2] Avionics Full-Duplex Switched Ethernet. https://en.wikipedia.org/wiki/Avionics_Full-Duplex_Switched_Ethernet.

[3] Bell number. http://mathworld.wolfram.com/BellNumber.html.

[4] S. A. Asghari, H. Taheri, H. Pedram, and O. Kaynak. Software-based control flow checking against transient faults in industrial environments. *IEEE Transactions on Industrial Informatics*, 10(1):481–490, Feb. 2014.

[5] M. Adnan, J.-L. Scharbarg, J. Ermont, and C. Fraboul. Model for worst case delay analysis of an AFDX network using timed automata. In *Proceedings of IEEE ETFA*, pages 1–4, Sep. 2010.

[6] M. Adnan, J.-L. Scharbarg, J. Ermont, and C. Fraboul. An improved timed automata model for computing exact worst-case delays of AFDX periodic flows. In *Proceedings of IEEE ETFA*, pages 1–4, Sep. 2011.

[7] M. Adnan, J.-L. Scharbarg, J. Ermont, and C. Fraboul. An improved timed automata approach for computing exact worst-case delays of AFDX sporadic flows. In *Proceedings of IEEE ETFA*, pages 1–8, Sep. 2012.

[8] M. Adnan, J.-L. Scharbarg, and C. Fraboul. Minimizing the search space for computing exact worst-case delays of AFDX periodic flows. In *Industrial Embedded Systems (SIES), 2011 6th IEEE International Symposium on*, pages 294–301, june 2011.

[9] A. Al Sheikh, O. Brun, M. Chéramy, and P.-E. Hladik. Optimal design of virtual links in AFDX networks. *Real-Time Systems*, pages 1–29, 2012.

[10] R. L. Alena, J. P. Ossenfort, K. I. Laws, A. Goforth, and F. Figueroa. Communications for integrated modular avionics. In *Proceedings of IEEE Aerospace Conference*, pages 1–18, Mar. 2007.

[11] R. Alur and D. L Dill. A theory of timed automata. *Theoretical computer science*, 126(2):183–235, 1994.

[12] M. Anand, S. Vestal, S. Dajani-Brown, and I. Lee. Formal modeling and analysis of the AFDX frame management design. In *Proceedings of IEEE ISORC*, pages 1–7, Apr. 2006.

[13] ARINC 664. *Aircraft Data Network Part 7 Avionics Full-Duplex Switched Ethernet Network*. AERONAUTICAL RADIO INC., 2009.

[14] ARINC 825-2. *General Standardization of CAN (Controller Area Network) Bus Protocol for Airborne Use*. AERONAUTICAL RADIO INC., 2011.

[15] SAE ARP4761. *Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment*. SAE International, Dec. 1996.

[16] J. Aweya. Technique for differential timing transfer over packet networks. *IEEE Transactions on Industrial Informatics*, 9(1):325–336, Feb. 2013.

[17] C. Baier and J.-P. Katoen. *Principles of model checking*. MIT press, 2008.

[18] A. Basu, S. Bensalem, M. Bozga, B. Delahaye, A. Legay, and E. Sifakis. Verification of an AFDX infrastructure using simulation and probabilities. In *Proceedings of the First International Conference on Runtime Verification*, pages 330–344, 2010.

[19] H. Bauer, J.-L. Scharbarg, and C. Fraboul. Applying and optimizing trajectory approach for performance evaluation of AFDX avionics network. In *Proceedings of IEEE ETFA*, pages 1–8, Sep. 2009.

[20] H. Bauer, J.-L. Scharbarg, and C. Fraboul. Improving the worst-case delay analysis of an AFDX network using an optimized trajectory approach. *IEEE Transactions on Industrial Informatics*, 6(4):521–533, Nov. 2010.

[21] H. Bauer, J.-L. Scharbarg, and C. Fraboul. Worst-case end-to-end delay analysis of an avionics AFDX network. In *Proceedings of DATE*, pages 1220–1224, Mar. 2010.

[22] H. Bauer, J.-L. Scharbarg, and C. Fraboul. Worst-case backlog evaluation of Avionics Switched Ethernet Networks with the trajectory approach. In *Proceedings of the 24th ECRTS*, pages 78–87, Jul. 2012.

[23] P. Bieber, F. Boniol, M. Boyer, E. Noulard, and C. Pagetti. New challenges for future avionic architectures. *AerospaceLab*, (4):1–10, 2012.

[24] M. Boyer and C. Fraboul. Tightening end to end delay upper bound for AFDX network calculus with rate latency FIFO servers using network calculus. In *Proceedings of IEEE WFCS*, pages 11–20, May 2008.

[25] M. Boyer, J. Migge, and N. Navet. A simple and efficient class of functions to model arrival curve of packetised flows. In *Proceedings of the 32nd IEEE RTSS*, 2011.

[26] M. Boyer, N. Navet, and M. Fumey. Experimental assessment of timing verification techniques for AFDX. In *Proceedings of ERTS, Toulouse, France*, 2012.

[27] M. Bozzano, A. Cimatti, J.-P. Katoen, P. Katsaros, K. Mokos, V. Y. Nguyen, T. Noll, B. Postma, and M. Roveri. Spacecraft early design validation using formal methods. *Reliability Engineering & System Safety*, 132:20 – 35, 2014.

[28] A. Burchard, J. Liebeherr, and S. D. Patek. A calculus for end-to-end statistical service guarantees. Technical Report CS-2001-19, Computer Science Department, University of Virginia, May 2002.

[29] G. Cena, I. C. Bertolotti, T. Hu, and A. Valenzano. Fixed-length payload encoding for low-jitter controller area network communication. *IEEE Transactions on Industrial Informatics*, 9(4):2155–2164, Nov. 2013.

[30] A. Cervin, D. Henriksson, B. Lincoln, J. Eker, and K. Arzen. How does control timing affect performance? Analysis and simulation of timing using Jitterbug and TrueTime. *IEEE Control Systems Magazine*, 23(3):16–30, Jun. 2003.

[31] C.-S. Chang. Stability, queue length, and delay of deterministic and stochastic queueing networks. *IEEE Transactions on Automatic Control*, 39(5):913–931, May 1994.

[32] C.-S. Chang, Y.-M. Chiu, and W. T. Song. On the performance of multiplexing independent regulated inputs. *ACM SIGMETRICS PER*, 29(1):184–193, 2001.

[33] H. Charara, J.-L. Scharbarg, J. Ermont, and C. Fraboul. Methods for bounding end-to-end delays on an AFDX network. In *Proceedings of the 18th ECRTS*, pages 193–202, 2006.

[34] F. Ciucu. *Scaling properties in the stochastic network calculus*. PhD thesis, University of Virginia, 2007.

[35] F. Ciucu, A. Burchard, and J. Liebeherr. Scaling properties of statistical end-to-end bounds in the network calculus. *IEEE Transactions on Information Theory*, 52(6):2300–2312, Jun. 2006.

[36] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen. *Evolutionary algorithms for solving multi-objective problems*. Springer, second edition, 2007.

[37] U.S. Nuclear Regulatory Commission. *Fault Tree Handbook*. 1998.

[38] Avionics Systems Standardization Committee. *Guide to Digital Interface Standards for Military Avionics Applications*. ASSC/110/6/2 issue 2, Sep. 2003.

[39] J. Craveiro, J. Rufino, C. Almeida, R. Covelo, and P. Venda. Embedded linux in a partitioned architecture for aerospace applications. In *Proceedings of AICCSA*, pages 132–138, May 2009.

[40] R. L. Cruz. A calculus for network delay, part I: Network elements in isolation. *IEEE Transactions on Information Theory*, 37(1):114 –131, Jan. 1991.

[41] R. L. Cruz. A calculus for network delay, part II: Network analysis. *IEEE Transactions on Information Theory*, 37(1):132 –141, Jan. 1991.

[42] G. B. Dantzig. *Linear programming and extensions*. Princeton University Press, 1998.

[43] M. Di Natale and A. L. Sangiovanni-Vincentelli. Moving from federated to integrated architectures in automotive: The role of standards, methods and tools. *Proceedings of the IEEE*, 98(4):603–620, Apr. 2010.

[44] D. M. Do, W. Gao, C. Song, and S. Tangaramvong. Dynamic analysis and reliability assessment of structures with uncertain-but-bounded parameters under stochastic process excitations. *Reliability Engineering & System Safety*, 132(0):46 – 59, 2014.

[45] F. Dobslaw, T. Zhang, and M. Gidlund. End-to-end reliability-aware scheduling for wireless sensor networks. *IEEE Transactions on Industrial Informatics*, pages 1–10, 2015.

[46] I. Dodd and I. Habli. Safety certification of airborne software: An empirical study. *Reliability Engineering & System Safety*, 98(1):7 – 23, 2012.

[47] A. D. Domínguez-García, J. G. Kassakian, J. E. Schindall, and J. J. Zinchuk. An integrated methodology for the dynamic performance and reliability evaluation of fault-tolerant systems. *Reliability Engineering & System Safety*, 93(11):1628 – 1649, 2008.

[48] K. Driscoll, B. Hall, P. Koopman, J. Ray, and M. DeWalt. Data network evaluation criteria report. Technical Report DOT/FAA/AR-09/27, 2009.

[49] S. P. Dwivedi. GCD computation of n integers. In *Proc. of RAECS*, pages 1–4, March 2014.

[50] M. Fidler. Extending the network calculus pay bursts only once principle to aggregate scheduling. In *Proceedings of the 2nd International Workshop on QoS-IP*, pages 19–34. Springer, 2003.

[51] F. Frances, C. Fraboul, and J. Grieu. Using network calculus to optimize the AFDX network. In *Proceedings of ERTS, Toulouse, France*, 2006.

[52] P. Frodyma and B. Waldmann. *ARINC 429 Specification Tutorial*. 2.1 edition, 2010.

[53] C. M. Fuchs. The evolution of avionics networks from ARINC 429 to AFDX. *Innovative Internet Technologies and Mobile Communications (IITM), and Aerospace Networks (AN)*, 65, 2012.

[54] R. Fuchsen. IMA nextgen: A new technology for the Scarlett program. *IEEE Aerospace and Electronic Systems Magazine*, 25(10):10–16, Oct. 2010.

[55] J.-P. Georges, T. Divoux, and E. Rondeau. Confronting the performances of a switched ethernet network with industrial constraints by using the network calculus. *International journal of communication systems*, 18(9):877–903, 2005.

[56] J.-P. Georges, T. Divoux, and E. Rondeau. Strict priority versus weighted fair queueing in switched Ethernet networks for time critical applications. In *Proceedings of the 19th IEEE IPDPS*, pages 141–141, 2005.

[57] M. Grenier, J. Goossens, and N. Navet. Near-optimal fixed priority preemptive scheduling of offset free systems. In *Proceedings of the 14th RTNS*, pages 35–42, 2006.

[58] L. Grillmayer. Determinism for Ethernet flows in industrial networks. *Network*, 73, 2014.

[59] J. J. Gutiérrez, J. C. Palencia, and M. G. Harbour. Response time analysis in AFDX networks with sub-virtual links and prioritized switches. *XV Jornadas de Tiempo Real*, 2012.

[60] J. J. Gutiérrez, J. C. Palencia, and M. G. Harbour. Holistic schedulability analysis for multipacket messages in AFDX networks. *Real-Time Systems*, 50(2):230–269, 2014.

[61] D. A. Gwaltney and J. M. Briscoe. Comparison of communication architectures for spacecraft modular avionics systems. *Marshall Space Flight Center, NASA/TM—2006-214431*, 2006.

[62] T. Hamza, J.-L. Scharbarg, and C. Fraboul. Priority assignment on an avionics switched Ethernet network (QoS AFDX). In *Proc. of WFCS*, pages 1–8, May 2014.

[63] B. W. Harrisand and B. J. Tran. Fiber optic AFDX for flight control systems. In *Proc. of IEEE AVFOP*, pages 15–17, Sep. 2012.

[64] X. He, Z. Wang, Y. Liu, and D. H. Zhou. Least-squares fault detection and diagnosis for networked sensing systems using a direct state estimation approach. *IEEE Transactions on Industrial Informatics*, 9(3):1670–1679, Aug. 2013.

[65] N. Hu, T. Lv, and N. Huang. Applying Trajectory approach for computing worst-case end-to-end delays on an AFDX network. *Procedia Engineering*, 15:2555–2560, 2011.

[66] Y. Hua and X. Liu. Scheduling design and analysis for end-to-end heterogeneous flows in an avionics network. In *Proceedings of IEEE INFOCOM*, pages 2417–2425, Apr. 2011.

[67] Y. Hua and X. Liu. Scheduling heterogeneous flows with delay-aware deduplication for avionics applications. *IEEE Transactions on Parallel and Distributed Systems*, 23(9):1790–1802, 2012.

[68] Y. Jiang and Y. Liu. *Stochastic network calculus*. Springer, 2008.

[69] M.-W. Kang, S.-W. Ha, and Y. Moon. Efficient data transmission scheme for real-time operation of mission computer. In *Proceedings of the 31st IEEE/AIAA DASC*, pages 10B1–1–10B1–10, Oct. 2012.

[70] A. Kavousi-Fard, M. A. Rostami, and T. Niknam. Reliability-oriented reconfiguration of vehicle-to-grid networks. *IEEE Transactions on Industrial Informatics*, 11(3):682–691, Jun. 2015.

[71] G. Kemayo, N. Benammar, F. Ridouard, H. Bauer, and P. Richard. Improving AFDX end-to-end delays analysis. In *Proc. of IEEE ETFA*, pages 1–8, Sep. 2015.

[72] G. Kemayo, F. Ridouard, H. Bauer, and P. Richard. Optimism due to serialization in the trajectory approach for switched Ethernet networks,". In *Proceedings of JRWRTC*, pages 13–16, 2013.

[73] G. Kesidis and T. Konstantopoulos. Worst-case performance of a buffer with independent shaped arrival processes. *IEEE Communications Letters*, 4(1):26–28, 2000.

[74] H. Kopetz and G. Grunsteidl. TTP - a time-triggered protocol for fault-tolerant real-time systems. In *Proceedings of the 23rd FTCS*, pages 524–533, Jun. 1993.

[75] J. Kurose. On computing per-session performance bounds in high-speed multi-hop computer networks. *SIGMETRICS PER*, 20(1):128–139, June 1992.

[76] M. Lauer, J. Ermont, F. Boniol, and C. Pagetti. Latency and freshness analysis on IMA systems. In *Proceedings of IEEE ETFA*, pages 1–8, 2011.

[77] M. Lauer, J. Ermont, C. Pagetti, and F. Boniol. Analyzing end-to-end functional delays on an IMA platform. In *Proceedings of the 4th ISoLA*, pages 243–257. Springer, Oct. 2010.

[78] J.-Y. Le Boudec and P. Thiran. *Network calculus: a theory of deterministic queuing systems for the Internet*, volume 2050. Springer, 2001.

[79] K. Lee. Performance bounds in communication networks with variable-rate links. In *ACM SIGCOMM Computer Communication Review*, volume 25, pages 126–136. ACM, 1995.

[80] L. Lenzini, E. Mingozzi, and G. Stea. Delay bounds for FIFO aggregates: a case study. *Computer Communications*, 28(3):287–299, 2005.

[81] C. Li, A. Burchard, and J. Liebeherr. A network calculus with effective bandwidth. *IEEE/ACM Transactions on Networking*, 15(6):1442–1453, Dec. 2007.

[82] J. Li, H. Guan, J. Yao, G. Zhu, and X. Liu. Performance enhancement and optimized analysis of the worst case end-to-end delay for AFDX networks. In *Proceedings of IEEE GreenCom*, pages 301 –310, Nov. 2012.

[83] M. Li, M. Lauer, G. Zhu, and Y. Savaria. Determinism enhancement of AFDX networks via frame insertion and Sub-Virtual Link aggregation. *IEEE Transactions on Industrial Informatics*, 10(3):1684–1695, Aug. 2014.

[84] M. Li, G. Zhu, M. Lauer, Y. Savaria, and J. Li. Incorporating performance analysis into reliability assessment for avionics full-duplex switched ethernet networks. *Reliability Engineering & System Safety*.

[85] M. Li, G. Zhu, Y. Savaria, and M. Lauer. Reliability enhancement of redundancy management in AFDX networks. *IEEE Transactions on Industrial Informatics*.

[86] X. Li. *Worst-case delay analysis of real-time switched Ethernet networks with flow local synchronization*. PhD thesis, 2013.

[87] X. Li, O. Cros, and L. George. The Trajectory approach for AFDX FIFO networks revisited and corrected. In *Proceedings of the 20th IEEE RTCSA*. LRT, Mar. 2014.

[88] X. Li, J.-L. Scharbarg, and C. Fraboul. Improving end-to-end delay upper bounds on an AFDX network by integrating offsets in worst-case analysis. In *Proceedings of IEEE ETFA*, pages 1–8, Sep. 2010.

[89] X. Li, J.-L. Scharbarg, and C. Fraboul. Analysis of the pessimism of the trajectory approach for upper bounding end-to-end delay of sporadic flows sharing a switched Ethernet network. In *Proceedings of RTNS*, pages 149–158. Citeseer, 2011.

[90] J. Magott and P. Skrobanek. Timing analysis of safety properties using fault trees with time dependencies and timed state-charts. *Reliability Engineering & System Safety*, 97(1):14 – 26, 2012.

[91] R. T. Marler and J. S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395, Apr. 2004.

[92] S. Martin and P. Minet. Schedulability analysis of flows scheduled with fifo: application to the expedited forwarding class. In *Proceedings of the 20th IPDPS*, pages 1–8, 2006.

[93] S. Martin, P. Minet, and L. George. End-to-end response time with fixed priority scheduling: trajectory approach versus holistic approach. *International Journal of Communication Systems*, 18(1):37–56, 2005.

[94] K. Michail, K. M. Deliparaschos, S. G. Tzafestas, and A. C. Zolotas. AI-based actuator/sensor fault detection with low computational cost for industrial applications. *IEEE Transactions on Control Systems Technology*, pages 1–9, 2015.

[95] I. Moir and A. Seabridge. *Aircraft systems: mechanical, electrical and avionics subsystems integration*. John Wiley & Sons, third edition, 2008.

[96] M. Nanda and S. Rao. A formal method approach to analyze the design of aircraft flight control systems. In *Proceedings of the 3rd Annual IEEE Systems Conference*, pages 64–69, Mar. 2009.

[97] P. M. Narendra and K. Fukunaga. A branch and bound algorithm for feature subset selection. *IEEE Transactions on Computers*, 26(9):917–922, Sep. 1977.

[98] W. Ni, I. B. Collings, R. Liu, and Z. Chen. Relay-assisted wireless communication systems in mining vehicle safety applications. *IEEE Transactions on Industrial Informatics*, 10(1):615–627, Feb. 2014.

[99] R. Obermaisser. *Time-triggered communication*. CRC Press, 2011.

[100] Department of Defense Military Handbook. *Digital Time Division Command/Response Multiplex Data Bus*. MIL-STD-1553B, Notice 4, 1996.

[101] M. Ohlin, D. Henriksson, and A. Cervin. *TrueTime 1.5 - Reference Manual*. Department of Automatic Control, Lund University, Jan. 2007.

[102] OpenFTA Official Website. *Formal Software Construction Ltd.* 2005.

[103] P. Pendyala and V. S. R. Pasupureddi. MIL-STD-1553+: Integrated remote terminal and bus controller at 100-mb/s data rate. In *Proceedings of IEEE ISCAS*, pages 1842–1845, May 2015.

[104] M. Plankensteiner. TTEthernet: A powerful network solution for all purposes. *Maxwell: Periodiek der Electrotechnische Vereeniging*, 13(3):30–33, 2010.

[105] F. Pozo, G. Rodriguez-Navas, H. Hansson, and W. Steiner. SMT-based synthesis of TTEthernet schedules: A performance study. In *Proceedings of the 10th IEEE International SIES*, pages 1–4. IEEE, 2015.

[106] S. Priya and B S. Rani. Design and development of AFDX transmitter schedular. *International Journal & Magazine of Engineering, Technology, Management and Research*, 2(6):760–767, 2015.

[107] N. Rastegar and E. Khorram. Relaxation of constraints in lexicographic multiobjective programming problems. *Optimization: A Journal of Mathematical Programming and Operations Research*, 64(10):2111–2129, 2015.

[108] J.-L. Scharbarg and C. Fraboul. Simulation for end-to-end delays distribution on a switched ethernet. In *Proceedings of IEEE ETFA*, pages 1092–1099, Sep. 2007.

[109] J.-L. Scharbarg, F. Ridouard, and C. Fraboul. A probabilistic analysis of end-to-end delays on an AFDX avionic network. *IEEE Transactions on Industrial Informatics*, 5(1):38–49, Feb. 2009.

[110] J. B. Schmitt, F. A. Zdarsky, and M. Fidler. Delay bounds under arbitrary multiplexing: When network calculus leaves you in the lurch.... In *Proceedings of IEEE INFOCOM*, pages 1669–1677, Apr. 2008.

[111] T. Schuster and D. Verma. Networking concepts comparison for avionics architecture. In *Proceedings of the 27th IEEE/AIAA DASC*, pages 1.D.1–1–1.D.1–11, Oct. 2008.

[112] K. Sentz and S. Ferson. Probabilistic bounding analysis in the quantification of margins and uncertainties. *Reliability Engineering & System Safety*, 96(9):1126 – 1136, 2011.

[113] M. Sghairi, A. De Bonneval, Y. Crouzet, J.-J. Aubert, and P. Brot. Architecture optimization based on incremental approach for airplane digital distributed flight control system. In *Proceedings of WCECS*, pages 13–20, Oct. 2008.

[114] D. P. Siewiorek and P. Narasimhan. Fault-tolerant architectures for space and avionics applications. In *First International Forum on Integrated System Health Engineering and Management in Aerospace*, pages 1–19, Nov. 2005.

[115] J. Sommer, S. Gunreben, F. Feller, M. Kohn, A. Mifdaoui, D. Sass, and J. Scharf. Ethernet - A survey on its fields of application. *IEEE Communications Surveys Tutorials*, 12(2):263–284, 2010.

[116] D. Song, X. Zeng, L. Ding, and Q. Hu. The design and implementation of the AFDX network simulation system. In *Proceedings of ICMT*, pages 1–4, Oct. 2010.

[117] C. R. Spitzer. *Digital avionics handbook*. CRC Press, 2007.

[118] D. Starobinski and M. Sidi. Stochastically bounded burstiness for communication networks. *IEEE Transactions on Information Theory*, 46(1):206–212, Jan. 2000.

[119] M. Tawk, G. Zhu, Y. Savaria, X. Liu, J. Li, and F. Hu. A tight end-to-end delay bound and scheduling optimization of an avionics AFDX network. In *Proceedings of the 30th IEEE/AIAA DASC*, pages 1–19, Oct. 2011.

[120] D. Trentin. Design and architecture of a hardware platform to support the development of an avionic network prototype. Master's thesis, École Polytechnique de Montréal, 2012.

[121] A. Vince. The greedy algorithm and coxeter matroids. *Journal of Algebraic Combinatorics*, 11(2):155–178, Mar. 2000.

[122] A. Vince. A framework for the greedy algorithm. *Discrete Applied Mathematics*, 121(1-3):247–260, Sep. 2002.

[123] L. F. Vismari and J. B. Camargo Junior. A safety assessment methodology applied to CNS/ATM-based air traffic control system. *Reliability Engineering & System Safety*, 96(7):727 – 738, 2011.

[124] M. Vojnovic and J.-Y. Le Boudec. Stochastic analysis of some expedited forwarding networks. In *Proceedings of IEEE INFOCOM*, volume 2, pages 1004–1013, 2002.

[125] M. Vojnovic and J.-Y. Le Boudec. Bounds for independent regulated inputs multiplexed in a service curve network element. *IEEE Transactions on Communications*, 51(5):735–740, may 2003.

[126] F. M. Waltz. An engineering approach: Hierarchical optimization criteria. *IEEE Transactions on Automatic Control*, 12(2):179–180, Apr. 1967.

[127] W. Weber, H. Tondok, and M. Bachmayer. Enhancing software safety by fault trees: experiences from an application to flight critical software. *Reliability Engineering & System Safety*, 89(1):57 – 70, 2005.

[128] M. B. Whiteside, S. T. Pinho, and P. Robinson. Stochastic failure modelling of unidirectional composite ply failure. *Reliability Engineering & System Safety*, 108(0):1 – 9, 2012.

[129] D. Wu and R. Negi. Effective capacity: a wireless link model for support of quality of service. *IEEE Transactions on Wireless Communications*, 2(4):630–643, Jul. 2003.

[130] J. Xia, W. Yuan, and R. Bai. Study on real-time performance of AFDX using OPNET. In *Proceedings of CASE*, pages 1–5, Jul. 2011.

[131] O. Yaron and M. Sidi. Performance and stability of communication networks via robust exponential bounds. *IEEE/ACM Transactions on Networking*, 1(3):372–385, Jun. 1993.

[132] Y. C. Yeh. Design considerations in Boeing 777 fly-by-wire computers. In *Proceedings of the 3rd IEEE International High-Assurance Systems Engineering Symposium*, pages 64–72, Nov. 1998.

[133] S.-E. Yoo, P. K. Chong, D. Kim, Y. Doh, M.-L. Pham, E. Choi, and J. Huh. Guaranteeing real-time services for industrial wireless sensor networks with IEEE 802.15.4. *IEEE Transactions on Industrial Electronics*, 57(11):3868–3876, Nov. 2010.

[134] J. Zhang, D. Li, and Y. Wu. Modelling and performance analysis of AFDX based on petri net. In *Proceedings of the 2nd ICFCC*, volume 2, pages V2–566–V2–570, May 2010.

[135] X. Zhang, X. Chen, L. Zhang, G. Xin, and T. Xu. End-to-end delay analysis of avionics full duplex switched ethernet with different flow scheduling scheme. In *Proceedings of IEEE ICCSNT*, volume 4, pages 2252–2258, 2011.

[136] T. Zheng. *Model Predictive Control*. Sciyo, Aug. 2010.

APPENDIX A TRUETIME

A.1 Overview of TrueTime

TrueTime is a Matlab/Simulink-based simulator, which can model data transmission using different network protocols and task execution in real-time kernels [30]. Furthermore, different scheduling policies, e.g., fixed-priority scheduling and earliest deadline first (EDF) scheduling, can be applied to initialize the kernels. Furthermore, the TrueTime network block supports multiple network models, e.g., CSMA/CD, Round Robin, TDMA [101]. Thus, TrueTime can be used as an experimental platform for research on real-time networking systems. The following introduction is based on the version of TrueTime 1.5.

The toolbox of TrueTime contains basic blocks, which are connected with ordinary Simulink blocks to form a real-time simulation system. The basic TrueTime simulation blocks and the interfaces are shown in Figure A.1.

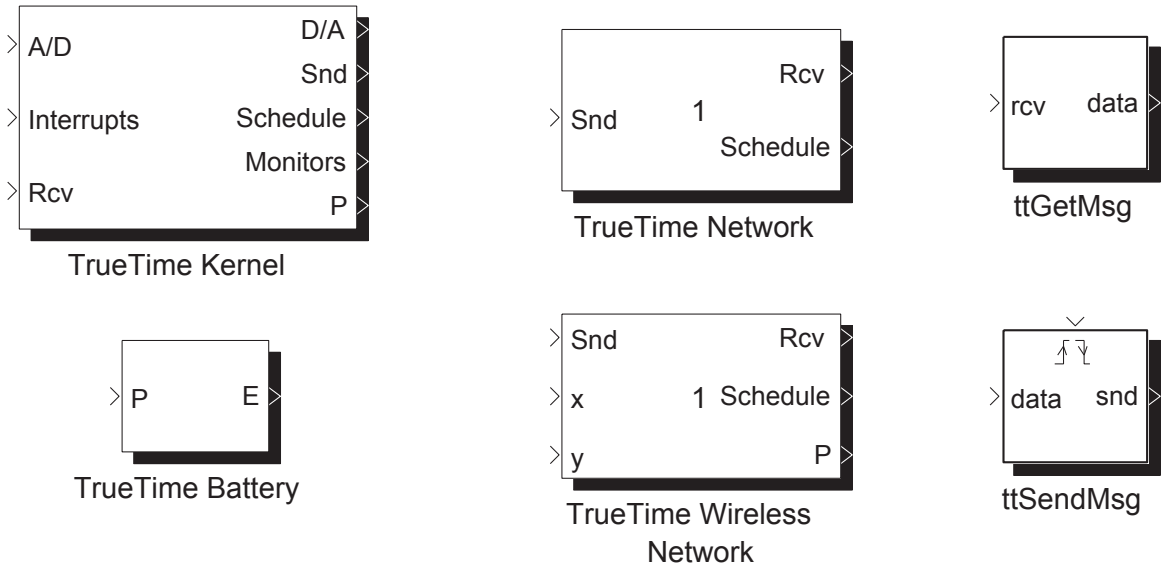


Figure A.1 Basic TrueTime simulation blocks.

Typically, a TrueTime kernel is responsible for logic control and data processing. The TrueTime network block realizes data distribution in a local network according to the selected network model. Wireless communication simulation is further provided by the use of TrueTime wireless network block, which supports the protocols of IEEE 802.11b/g and IEEE 802.15.4. The battery block is designed to provide a power supply based on a simple inte-

grator model and thus it can be recharged. Besides, two more standalone network blocks are provided to offer more options for message delivery. For the simulation system in this thesis, only TrueTime kernel block and TrueTime network block are employed.

A.2 TrueTime Kernel Block

TrueTime kernel block requires to be initialized before running the simulation. The initialization file can be either a C++ file or a Matlab M-file, both of which are support by TrueTime. As shown in Figure A.2, the initialization is specified as the function with the name of "SubVL1_Init". The kernel initial argument is assigned in the block dialogue. If the kernel is battery-powered, the corresponding check box is enabled. The other two parameters, clock drift and clock offset, can be configured to imitate special conditions.

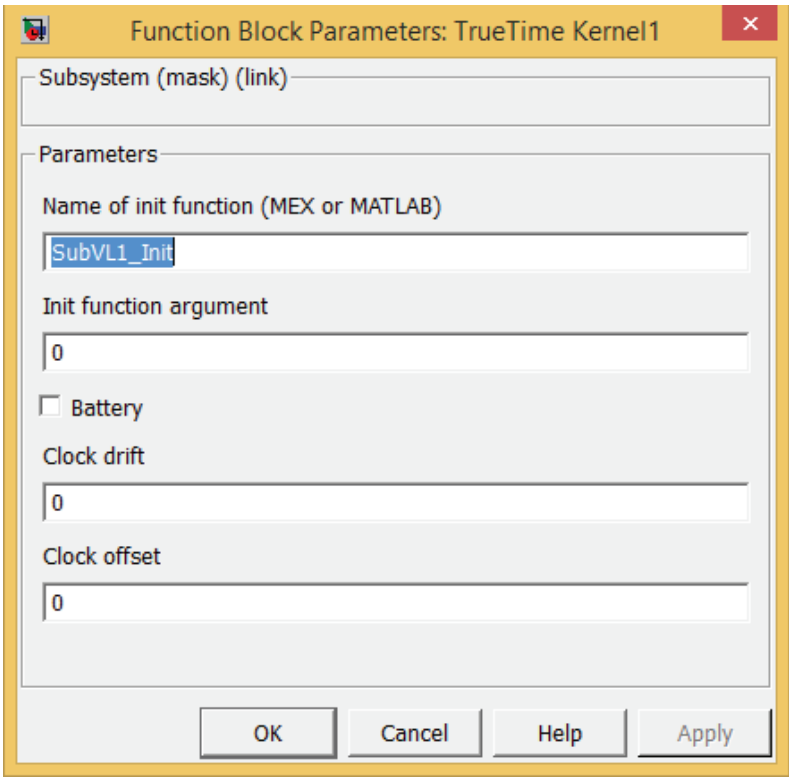


Figure A.2 The TrueTime kernel parameters.

A.3 TrueTime Network Block

TrueTime network block simulates data transmission in a local network. The transferred frames contain diverse information, e.g., source, destination, data length, priority, etc. The

network block supports six models: CSMA/CD, CSMA/AMP, RR, FDMA, TDMA and Switched Ethernet. In this thesis, the network block associated with RR is used to simulate the Sub-VL aggregation. The Switched Ethernet configuration is suitable for AFDX switches, in which FIFO scheduling is adopted. Please note that only packet-level simulation is supported by TrueTime network block.

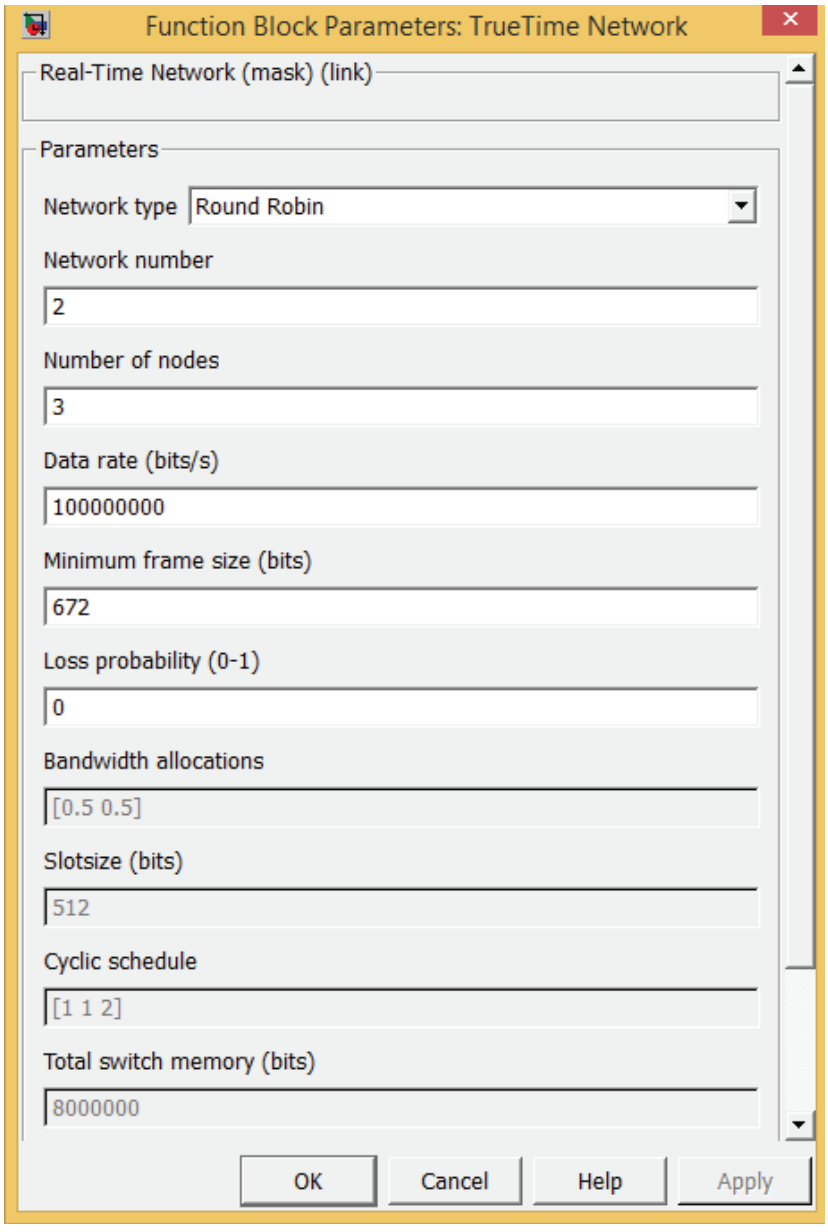


Figure A.3 The TrueTime network parameters.

Multiple parameters can be assigned in the block dialogue of TrueTime network including the data rate, the minimum frame size, the loss probability, etc. All of these parameters

should be decided according to the practical applications and the focus of the simulation.

A.4 TrueTime Commands

TrueTime provides diverse commands, which can be used to set up the simulation model. According to the functionality, these commands can be classified into one or more of the following categories: initialization script, task code function and interrupt handler code function. For example, the command of "ttInitKernel" is used to initialize the kernel and it belongs to the initialization script category. Another command, "ttGetData", can be applied for both task code function and interrupt handler code function. For more details about commands, please refer to [101].

A.5 TrueTime Modeling of AFDX

Tow essential elements in the simulation of AFDX networks are ESs and switches. Normally, a TrueTime kernel block with an proper initialization file is able to model one ES to execute different tasks and perform data delivery to the switches. However, an ES with some complex mechanism, such as a Sub-VL aggregation management, cannot be imitated by one single TrueTime kernel and it requires the cooperation of multiple blocks.

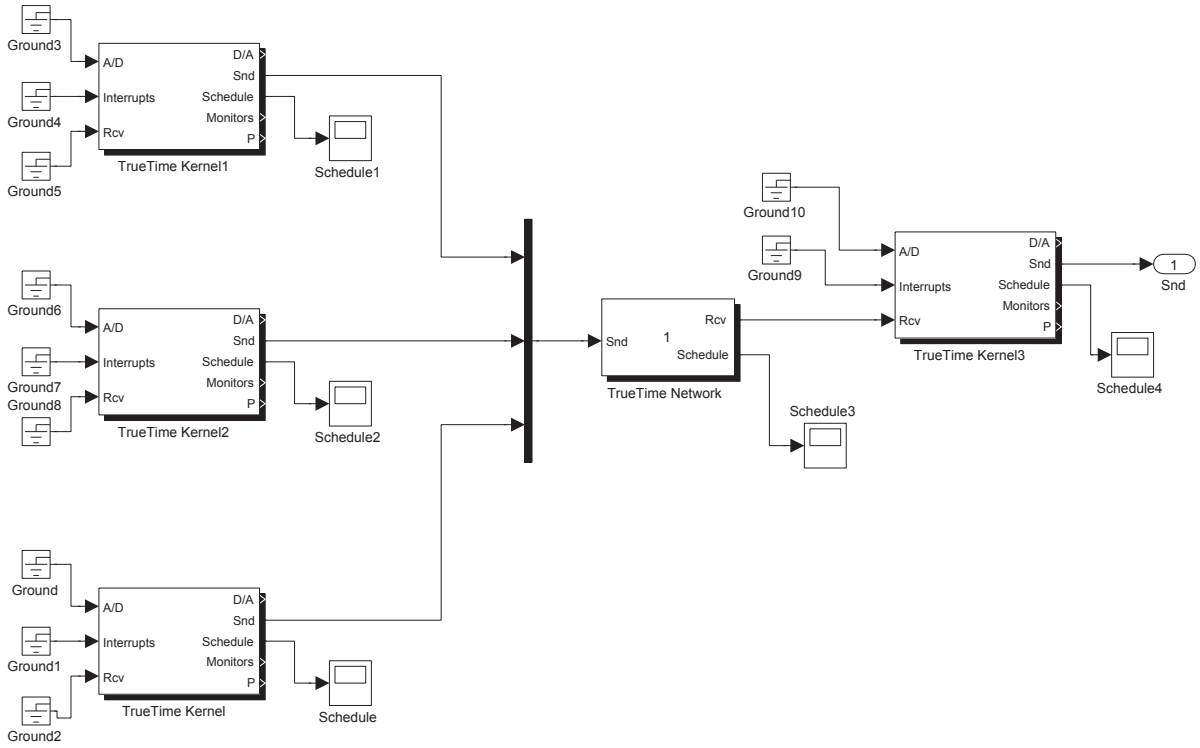


Figure A.4 Modeling of an ES with Sub-VL aggregation by TrueTime.

As shown in Figure A.4, a simulation of an ES with Sub-VL aggregation is carried out. In the considered example, there are three Sub-VLs. Each Sub-VL is modeled by a TrueTime kernel and the VL FIFO as shown in Figure 4.5 is implemented by a TrueTime Network associated with RR scheduling. Furthermore, another TrueTime kernel is applied to perform the VL regulation, the frame insertion and the VL scheduling. Then all the frames of VLs are forwarded into the connected switches.

Generally, the function of a switch is implemented by the combination of a TrueTime kernel and a TrueTime Network. As shown in Figure A.5, a switch model is given, which includes two inputs and one output. In this model, a TrueTime Network block is applied to receive the frames with proper configurations and a TrueTime kernel is used to realize traffic filtering and traffic policing. The processed frames are sent to a further destination by the TrueTime kernel. The number of either inputs or outputs can be adjusted according to the practical requirements, in which case the initialization file of the TrueTime kernel and the configurations of the TrueTime Network should also be modified accordingly.

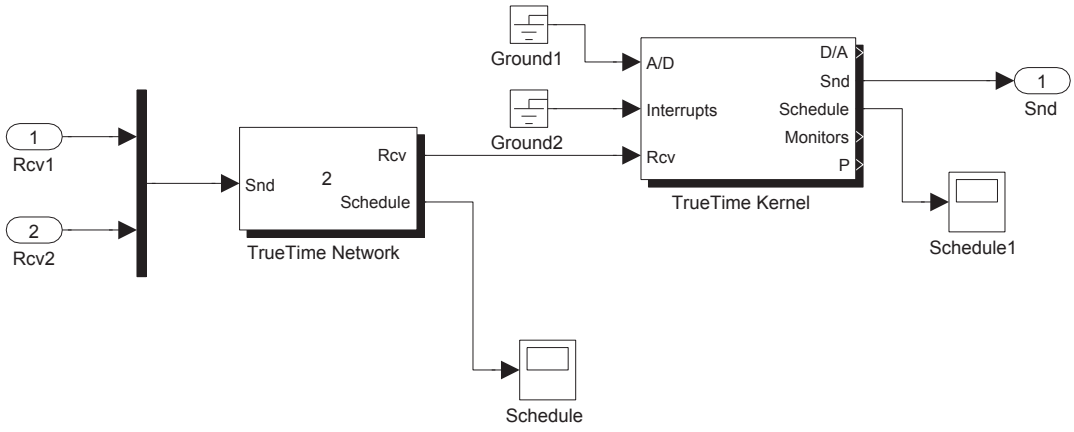


Figure A.5 Modeling of an AFDX switch by TrueTime.